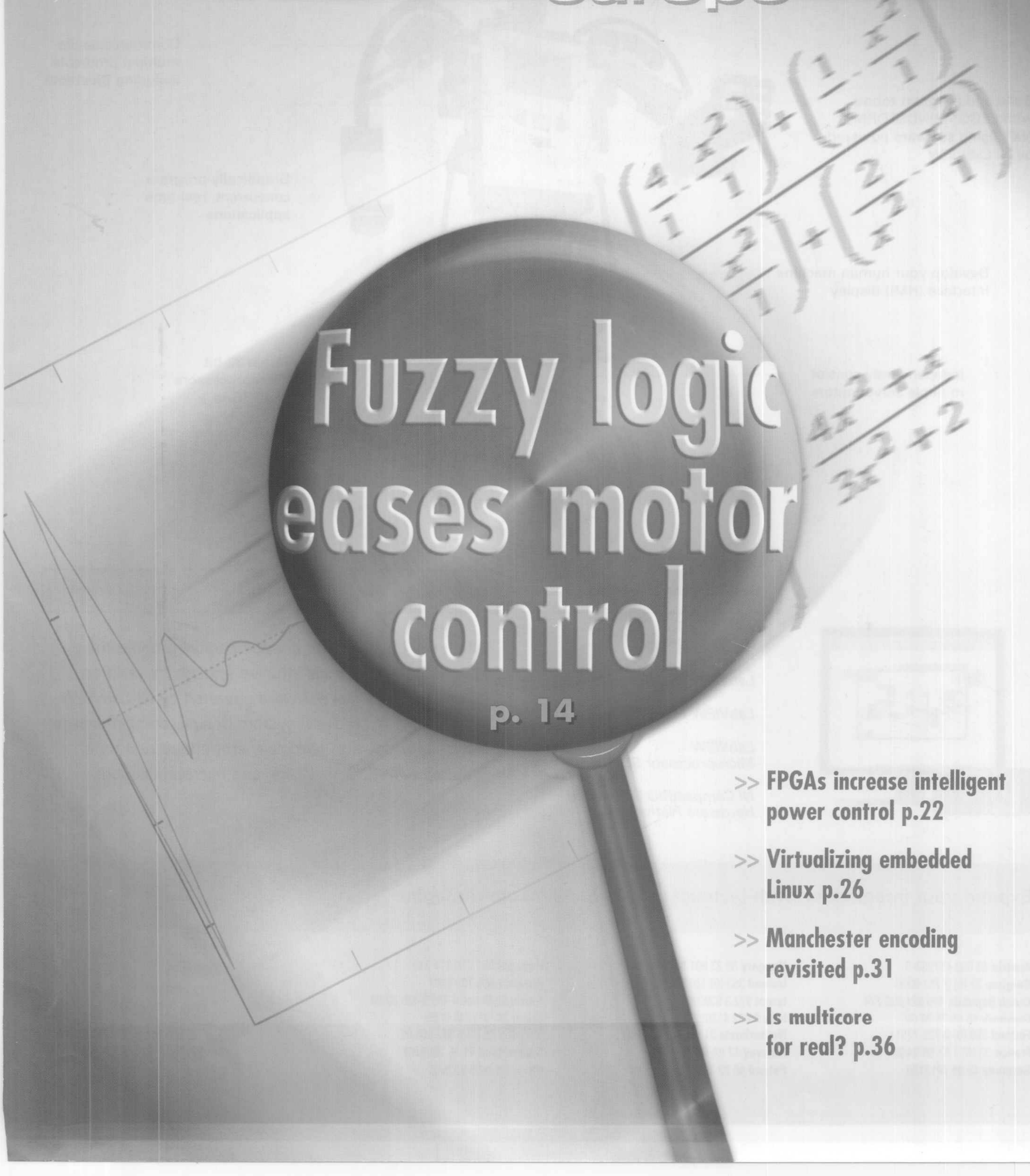
United Business Media

March 2008

Embedded Systems Design europe



Fuzzy logic eases motor control

p. 14

THE EUROPEAN JOURNAL FOR THE EMBEDDED SYSTEMS DESIGNER

Embedded Systems Design europe

www.embedded.com/europe

MARCH 2008 VOLUME 12, NUMBER 88

Software security – a matter of life and death

Last month in this column we were anticipating an interesting visit to Embedded World in Nuremberg and the event turned out to meet all our expectations. The exhibition set new records with 675 exhibitors – up 14 percent – and 17,341 trade visitors from all over the world – up 27 percent. Particularly impressive was the rise of 56 percent in international attendance. Engineers thronged the exhibition aisles and conference centre to be informed and educated but it was a bonus to be entertained.

David Kleidermacher, CTO of Green Hills Software, did exactly that in his executive summit keynote which looked at how healthy software is, especially in safety critical systems.

Kleidermacher felt the general public think bad software security is only the subject of futuristic adventures and used the Bruce Willis movie colloquially known as Die Hard 4 but officially titled Live Free or Die Hard as an example.

In the film cyber terrorists were able to hack into Washington DC's traffic control system and turn all the lights green causing chaos. Kleidermacher was watching the movie with his wife who, taking advantage of having an expert in the family, posed the question: Can this really happen? It turns out that the proposition was not that far fetched with a number of examples on record including a hack of the LA traffic light system and reports that the U.S. Accountability Office found that the nations air traffic control system was wide open to hacking while in a separate incident a PC crash put 800 planes in jeopardy with pilots having to use cell phones to communicate with each other.

In another scene in the film the good guys persuaded an operator in a remote call center to switch on the ignition of a car for which they didn't have the keys. That might seem far fetched said Kleidermacher but maybe not, as late last year General Motors announced an improved version of its OnStar telematics system that could be used to shut your car's engine off remotely. "Be afraid, be very afraid," warned Kleidermacher.

What about the power grid. In the film Willis watches from a hill at night as sections of the Washington DC landscape go black in turn as the hackers turn the power off. Another exaggeration, no said Kleidermacher, in 2004 a software bug contributed to a blackout of northern eastern U.S. As a result of the outage, New York City reported 60 serious fires and 3,000 fire calls, 800 elevator rescues, and 80,000 calls to 911.

As the movie comes to its climax a flight of F-35 jets are scrambled and one jet's command and control communications system is hacked into and it is ordered to attack Willis destroying much of the freeway in the process.

Not another far-fetched plot that has actually happened? At last Kleidermacher gets his pay off. The F-35 uses Green Hills' INTEGRITY operating system for its mission and display systems, its communication/navigation/identification systems and crypto engine. It is based on EAL6+ certification – Windows and Linux are only EAL4 level – and this higher level is the security we need to stop our weapons being turned on ourselves, explained Kleidermacher. But this level could also be used selectively in commercial applications using multicores and hypervisors to improve security.

Colin Holland is editor of Embedded Systems Design Europe. *He can be contacted at cholland@cmp.com.*

Editor
Colin Holland
Tel +44 (0) 208 319 1324
email: chollandse18@mac.com
Editorial address: PO Box 32444, London SE18 3ZP

Advertising Production Manager
Lydia Gijsegom & Maggie Fleerackers
Tel +32 (0) 2 740 00 50
email: production@eetimes.be

Circulation
Luc Desimpel
Tel +32 (0) 2 740 0055
email: luc.desimpel@eetimes.be

Art Manager
Jean-Paul Speliers
Tel +32 (0)2 740 0052
email: jean-paul.speliers@eetimes.be

Finance
Ricardo Pinto Ferreira
Tel +32 (0)2 740 0051
email: ricardo.pintoferreira@eetimes.be

Advertisement contacts
see page 43

Editorial Director
Jean-Pierre Joosting
Tel +32 (0)2 740 0056
email: jean-pierre.joosting@eetimes.be

Publisher
Andre Rousselot
Tel +32 (0)2 740 0053
email: aroussel@eetimes.be

European Business Press
144 Avenue Eugène Plasky
1030 Brussels - Belgium
Tel: +32 (0)2 740 00 50
Fax: +32 (0)2 740 00 59
Email: info@eetimes.be
www.embedded.com/europe
VAT Registration: BE 461.357.437
RPM: Brussels
Company Number: 0461357437

techinsights
inspiring the creators of technology

Techinsights – a division of
United Business Media, LLC
Paul Miller
President, Chief Executive Officer

• Embedded Systems Design Europe is available free to qualified readers and to others for an annual subscription of €150. Refunds on cancelled subscriptions will only be provided at the Publisher's discretion. Subscription queries tel: +32 2 740 00 50

• Printed by Perka.

United Business Media

Distributors to increase embedded focus

The two top electronic component distributors are both addressing the importance of the embedded systems sector by launching specialist divisions.

Arrow is to increase its European focus on integrated system-level technologies with launch of Advanced Embedded Solutions (AES).

AES is designed to provide a European focus on platform-level technologies, AES covers integrated solutions including displays, industrial computer products, solid state storage and power supplies, as well as wireless modules for machine-to-machine (M2M) communication and operating systems.

The new organization is headed by Chris McAneny as European embedded solutions group leader. McAneny is responsible for AES teams in Arrow organisations throughout Northern, Central and Southern Europe.

"Pressure on in-house engineering resource is affecting the 'make versus buy' balance. Increasingly, customers want to focus design effort around core competencies while buying in solutions that will allow them to minimise time-to-market. AES brings together the technologies and resources that address customer demand for the integrated solutions, while Arrow's component teams continue to deliver the products and technical support for customers' own circuit designs," said McAneny.

Among the technologies within the AES portfolio are LCDs, single board computers, system-on-module (SOM) solutions, solid stage storage, embedded operating systems, AC/DC power supplies, DC/DC converters and Bluetooth, ZigBee, Wireless LAN and ISM radio modules. A host of related and complementary technologies and services including wireless antennae, display controllers, configuration services and training will also be handled through the AES channel.

Avnet Electronics Marketing EMEA's new channel for embedded systems, networking and display products, is called Avnet Embedded.

It will initially do business in Germany, Italy, Spain and Eastern Europe, but over time extend its activities to other major European markets. Avnet Embedded will be led by Alessandro Brazzoni, formerly manager of the embedded division at Avnet Memec, now director of Avnet embedded sales & operations.

Avnet Embedded will represent a number of manufacturing partners, including embedded boards from AAEON, Advantech, aValue, Protech, TQ Components and VIA Technologies; networking products from Atop, Delta Design, Digi International, Lancom, Maxstream, Tibbo, and Volktek; displays from Litemax, Promate (AUO), Sharp, Varitronix, and Xiamen. Not all of the listed manufacturers will be available in all countries of operation from the beginning.

Patrick Zammit, president of Avnet Electronics Marketing EMEA, said "The market for embedded system products beyond the pure chip level is increasing rapidly. Many customers ask for solutions of higher integration, to reduce time to market and development costs, and therefore are starting to work with finished and semi-finished solutions. By providing embedded systems products, we just naturally extend our offering to meet some of our customers needs."

Brazzoni added, "The strength of the Avnet speedboats is the excellent customer relationship combined with technical expertise and the focus on a limited number of franchises. Avnet Embedded entertains a completely new and different line card with complex products that need separate technical support and, to a large extent, a different logistic handling. Offering board level solutions and displays through a new product group enables Avnet – through Avnet Embedded – to leverage the excellent market penetration of Avnet EM-EMEA while creating a specific technical focus for embedded products."

Kontron, Quanta join forces

Notebook computer manufacturer Quanta Computer, Inc. (Taipei, Taiwan) has agreed with embedded computer vendor Kontron AG (Eching, Germany) to take a minority stake in Kontron's manufacturing subsidiary Kontron Asia, Inc. With the move, Kontron hopes to benefit from Quanta's manufacturing expertise and improve its productivity.

The agreement will see Quanta buy 21 percent of Kontron Asia's shares. The closing of the transaction is expected before end of March according to Kontron. The move effectively creates a joint venture of both companies. Kontron Asia's largest asset is a manufacturing site in Penang, Malaysia.

Kontron will maintain the operative leadership in the joint venture, while both companies hope to benefit equally from the move: Kontron aims at achieving the same conditions in material procurement as the much larger Taiwanese partner. In addition, Kontron plans to benefit from economies of scale through the partnership. Quanta is said to be the world's largest notebook computer manufacturing services provider, producing about 30 percent of the world's notebook motherboards. For Quanta, the move could translate in a diversification of its production capacities for industrial markets.

■ Kontron is also in the process of buying Thales Computers S.A., a company which expects sales of about €20 million in its present financial year. The purchase is subject to approval by the French government with deal expected to be finalized in March.

For land-based or airborne applications, the military needs COTS embedded sub-systems that deliver the optimum combination of performance, reliability, cost and rapid time-to-market. And Aitech has delivered just that since 1983!

We're still in front... Aitech got a big jump on everyone way back when…and we've kept well ahead of the pack in developing advanced embedded sub-systems based on leading edge commercial technologies adapted to harsh environment applications.

We don't compete with you... A lot of board level companies claim to be systems integrators. We don't. We leave the systems integration to the companies that do it best: our customers...companies just like yours!

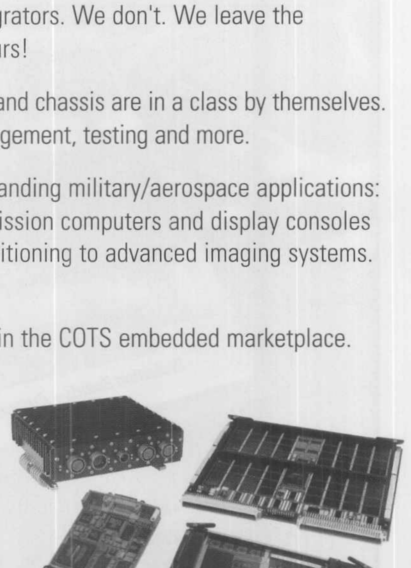
We have what it takes... Our VME and CompactPCI SBCs, PMCs, memory, I/O, graphics and chassis are in a class by themselves. And they're backed by world class software and technical support, program and lifecycle management, testing and more.

We've been there... Aitech sub-systems are in widespread use in the world's most demanding military/aerospace applications: from radar, missile and weapons control to mission-critical data voice sub-systems…from mission computers and display consoles to ECM, sight head stabilization and counter measure control…from power control and conditioning to advanced imaging systems. And that's just for starters.

Don't' take our word for it... Compare Aitech's offerings and expertise to any company in the COTS embedded marketplace. Land or air. Or for that matter, sea or space. Come to talk to us.

Aitech

Aitech Systems, Ltd.
4 Maskit Street
Herzlia, Israel 46120
email: sales@rugged.com
Tel: +972 (9) 9600600
Fax: +972 (9) 9544315
www.rugged.com



Coverity raises $22m as European business booms

Coverity Inc. has raised its first round of external funding with a $22 million investment by Foundation Capital and Benchmark Capital. The company has seen five consecutive years of positive cash flow growth and also grew in Europe by 300 percent in 2007.

It will use this investment for acquisitions, internal product development, and further expansion into international markets. The company says its Prevent static analysis solution is now used by hundreds of thousands of developers.

The company's expanded product portfolio will allow Coverity to establish a new presence in targeted segments of the market for application lifecycle management tools, a sector which Gartner has predicted will grow in European to more than $2.9 billion in 2011. Steve Machernis, European managing director at Coverity said, "Coverity is hiring aggressively throughout Europe while cultivating our network of partners."

Coverity has recently augmented its board of directors by adding Tony Zingale, who was formerly president and CEO of Mercury Interactive; Aki Fujimura, who is currently chairman/CEO of D2S and formerly was CTO of Cadence Design Systems; Bruce Dunlevie of Benchmark Capital; Paul Holland of Foundation Capital; and Mike Schuh of Foundation Capital who was formerly CEO at Intrinsa, and vice president of sales at Clarify and Cadence.

Help is on hand for Europe's industrial control developers

EE Times Europe has launched its first DesignLine – Industrial Control DesignLine Europe.

The site offers product news, in-depth technical papers and how-to articles, with a European perspective.

The website supports engineers during the detailed design phase of their project, when they need to delve deeply into the software and hardware details of their designs. During this phase, engineers need access to many different types of information, ranging from product specifications to how-to instructions and trouble-shooting hints.

The information on the site is focused on this phase of the product development cycle as well as on the specific requirements of industrial electronics – and all this with an unmistakably European flavor which it will develop over time.

There is also an editor's blogs, commenting on latest trends and events in the industry.
See: **www.industrial-europe.com**

Milestones in embedded systems

How far we've come? Jack Ganssle and the Embedded Systems Design staff have created a list of some of the important happenings in the history of embedded systems. It can be viewed at **www.embedded.com/timeline** and we invite comments and suggestions. Are we missing any events or inventions that are pertinent to embedded systems? Do you have any good photos you're willing to share? Send your suggestions and photos to srambo@techinsights.com or log on to the forum on the website and post your comments.

Microsoft is recruiting for embedded center in Aachen

Microsoft Corp. has launched its first Microsoft Embedded Systems Development Centre (MESDC) which will be based in Aachen, Germany.

The MESDC is part of Microsoft's effort to expand regional development centers in Denmark, France, Ireland, Serbia, the U.K. and other countries across Europe.

Located within the European Microsoft Innovation Centre (EMIC) in Aachen which was set up in 2003, the company says the MESDC is a significant part of the $75 million global R&D investment that the Windows Embedded Business is making in Europe this fiscal year.

Microsoft is recruiting embedded systems engineers for the MESDC and the aim is to bring up to 15 engineers on board by the end of 2008 to work closely with the U.S. Windows Embedded product development team based in Redmond, Washington.

The MESDC will support global product R&D, development of Microsoft's embedded operating systems, and accelerate collaboration between the U.S.-based Microsoft product groups and their counterparts in Europe.

The MESDC will also support the needs of the active Windows Embedded customer and partner ecosystem in Europe by engaging with select members of this group to showcase high-visibility embedded systems projects that accelerate embedded development.

"The MESDC will form an integral part of helping us recruit outstanding embedded systems engineers across Europe to perform product R&D, and the MESDC located in the EMIC in Aachen provides a very conducive environment for these engineers to spearhead incubation projects and accelerate technology transfer and collaborative efforts within Microsoft," according to John Lefor, the director of the European Microsoft Innovation Center in Aachen.

Kevin Dallas, general manager of the Windows Embedded Business at Microsoft, added, "Due to the immense engineering talent pool and high concentration of enterprise customers and key Windows Embedded partners in the region, we see tremendous growth opportunities for the Windows Embedded Business in Europe.

Duo work on smaller form factor

Congatec (Deggendorf, Germany) and SECO (Arezzo,Italy) have jointly created the Qseven form factor for embedded computing using the latest low power chipsets such as Intel's Menlow platform. Targeting next generation embedded processors built using 45nm technology, Qseven format is a board measuring 70 x 70mm and a maximum power consumption of around 12W.

Where heat dissipation is an issue, a thermal cooling interface has been defined to help transfer any heat generated to a cooling solution. It will also provide connectivity through industry standard interfaces, including: 4x PCI Express; 2x SATA; 6x USB 2.0; 1x 1000BaseT Ethernet; 2x SDIO 8 bit; LVDS 2x 24 bit; DVO/SDVO (shared); video input port (VIP); high definition audio (HDA); I^{2}C bus; and low pin count bus. By leveraging the Mobile PCI Express Module connector format, Qseven offers three different connector heights, from 4.3mm to 7.8mm. Through this configuration it is hoped the Qseven platform will be as simple to integrate as a DIMM memory module.

The companies plan to release a general specification by the end of April, with a full design guide available by May 2008.

European designers to win cash for green designs

Freescale Semiconductor used the Embedded World conference to open registration in Europe for its first worldwide design contest. Top regional winners will be invited to compete in the Grand FTF Design Challenge, competing against other first place teams to win a total of $61,000 in cash prizes.

The Freescale Technology Forum (FTF) Design Challenge Europe is focused on the development of sustainable applications to encourage engineers to test the boundaries of green embedded systems design.

"Energy management and other green issues are becoming major considerations in virtually every purchase decision: consumer or corporate," said Steve Wainwright, vice president of sales and marketing & general manager, Freescale EMEA.

Embedded systems engineers and students are challenged to create innovative, green electronic designs using a select group of Freescale solutions. Once registered, participants receive a kit with hardware and software development tools to start on their designs, submitting final design papers by May 23. The judging process will coincide with FTF Paris, which will be held October 7 – 8, where top regional winners will be recognized and awarded cash prizes.

Freescale is hosting Design Challenges in the Americas, Japan, Europe, China and India. To be eligible to participate in the Europe Design Challenge, participants must be 18 years or older and live in Europe including Israel. First place regional winners will receive $10000, in addition to being invited to participate in the Grand FTF Design Challenge. Second place winners receive $5000 and third place $2000. Additional information including entry requirements and the official rules are available at www.freescale.com/designchallenge.


YOUR FULL SPECTRUM SOLUTION
300,000 PARTS IN STOCK
375+ MANUFACTURERS
FREE SHIPPING ON ORDERS OVER €150!*
Digi-Key® CORPORATION
Quality Electronic Components, Superior Service
More than 300,000 products stocked and ready to ship today!**
Digi-Key Corporation purchases all product directly from its original manufacturer.
*A shipping charge of €18.00 will be billed on all orders of less than €150.00. All orders are shipped via UPS for delivery within 1-3 days (dependent on final destination). All prices are in Euro and British pound sterling and include duties.
**New product added daily.
© 2008 Digi-Key Corporation
Austria at.digikey.com 0800.291.395
Belgium be.digikey.com 0800.78.962
Denmark dk.digikey.com 80.88.13.67
Finland fi.digikey.com 0800.115.281
France fr.digikey.com 0800.161.113
Germany de.digikey.com 0800.1.800.125
Greece gr.digikey.com 00800.3122.3456
Ireland ie.digikey.com 1.800.552.170
Italy it.digikey.com 800.786.310
Luxembourg lu.digikey.com 800.24.398
Netherlands nl.digikey.com 0800.020.3710
Norway no.digikey.com 800.15.286
Portugal pt.digikey.com 800.831.521
Spain es.digikey.com 900.983.183
United Kingdom uk.digikey.com 0.800.587.0991
0.800.904.7786
Or visit www.digikey.com for ordering information.
701 Brooks Ave. South • Thief River Falls, MN 56701 • USA

Europe invests in Java for multicore systems

The Open Group is working with a consortium of European real-time technology developers, industrial manufacturers and research organizations to develop a new framework for Java-based real-time applications on modern parallel processor systems.

Supported by the European Commission, the Java Environment for Parallel Real-time Development (JEOPARD) project is investing over €3.3 million on an advanced framework for real-time Java running on multicore and parallel systems.

This platform-independent framework will maintain the robust reliability essential for safety and mission critical applications, while using the additional processing power available from the latest parallel platforms.

Led by The Open Group (Reading, England), the JEOPARD consortium includes four universities and research institutes: University of York (England), Vienna University of Technology (Austria), FZI (Research Center for Information Technologies at the University of Karlsruhe, Germany) and the Technical University of Cluj-Napoca (Romania); three industrial manufacturers: EADS NV (Schiphol-Rijk, Netherlands), RadioLabs (Rome, Italy) and SkySoft (Lisbon, Portugal); and two embedded systems technology suppliers: Aicas (Karlsruhe, Germany) and Sysgo (Klein-Winternheim, Germany).

The strategic objective of the JEOPARD project is to provide the tools for platform-independent development of predictable systems that make use of symmetric multiprocessing (SMP) multicore platforms. These tools will enhance software productivity and reusability by extending processor technology already established on desktop systems for the specific needs of multicore embedded systems.

The project will actively contribute to standards required for the development of portable software in this domain, such as the Real-time Specification for Java (RTSJ).

In addition, the JEOPARD project will develop a platform-independent software development interface for realtime multicore systems. The interface will be based on existing technologies, including the Real-Time Specification for Java (JSR 1 and JSR 282) and Safety-Critical Java (JSR 302).

These technologies currently provide a strong foundation for the development of complex and highly reliable real-time systems, but they do not yet provide support for multicore systems. Even more challenging, some of the technologies can not address more than one processor at a time, making it impossible to develop applications that scale with the number of processors available on current and future multicore systems.

"The JEOPARD consortium partners are experts in all layers of a multicore embedded system, from the processor architecture to the Java virtual machine and the C and Java API layers," said Scott Hansen, director of European Projects at The Open Group. "We're confident the project will solve the critical issues in each of these layers, thereby providing a powerful interface for the applications developer at the highest level."

"Embedded systems developers require a stable platform based on recognized standards if they are to invest in new applications that fully exploit multicore systems," said Fridtjof Siebert, CTO of Aicas and technical director for the consortium. "The JEOPARD project will provide an evolutionary path that maintains the advantages of Java such as portability, interoperability, object orientated design, while enabling real-time systems developers to achieve higher performance levels and scalability with increasingly sophisticated multicore platforms."

Curtiss-Wright buys Pentland Systems

Curtiss-Wright Controls Embedded Computing has acquired Pentland Systems Ltd. (Livingston, Scotland), for $2.3 million. Pentland was founded in 1990 and its sales in 2008 are expected to be approximately $3.6 million. The deal should enhance Curtiss-Wright's sensor processing product portfolio adding RF/IF signal acquisition, analog, digital I/O and synchro/resolver products for radar, software defined radio and signal intelligence applications.

Pentland's products and technology will be organized under Curtiss-Wright Controls Embedded Computing's Modular Solutions group, and will be managed by Rob Hoyecki, director of the digital signal processing division.

"Pentland is a leading supplier of rugged high performance analog acquisition and reconstruction products for sonar, radar, signal intelligence and software defined radio applications in the aerospace and defense market," said Tom Quinly, president of Curtiss-Wright Controls Embedded Computing. "The addition of another incredibly talented and complementary technical team fits our strategy of continuing to expand our portfolio to offer the broadest and most tightly integrated rugged deployed COTS boards and subsystems available."

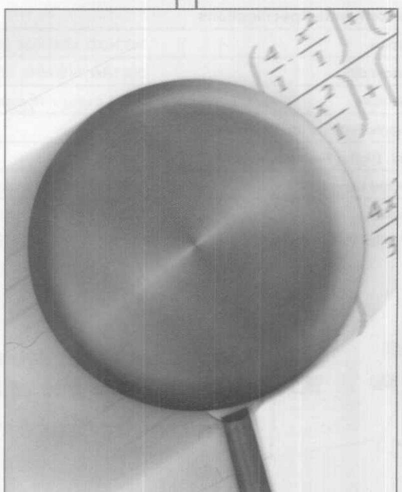
Hoyecki added, "Their approach, which tightly couples the analog front-end with FPGA technology is highly desired by our customers who require a defined signal bandwidth to be delivered at the application level." The acquisition strengthens Curtiss-Wright's presence in Europe, adding to their existing video and integration business unit in Letchworth, which became part of the group in 2004 with the purchase of Primagraphics for $21million.

cover feature

Designing DSP-based motor control using fuzzy logic

BY BYRON MILLER

The increased use of variable-speed drive motors to reduce energy consumption will require a shift from PID controllers to systems based on fuzzy logic algorithms to simplify design, reduce development time, and eliminate complex math formulas.



Variable-speed drive (VSD) motors provide hope for greatly reducing energy consumption and reliance on foreign fuels. In one approach, digital signal processors (DSPs) are being used to create a new generation of VSD-based controllers for motors such as brushless direct current (BLDC) motors.

However, these motors present challenges. Controlling motor speed on a BLDC motor is complicated when using traditional proportional, integral, and differential (PID) controllers because they rely on a complex mathematical model and are computationally intensive. An alternative approach is to use fuzzy logic (FL) algorithms to eliminate the need for complex math formulas and provide an easy-to-understand solution. FL motor con-

The BLDC controller block diagram.

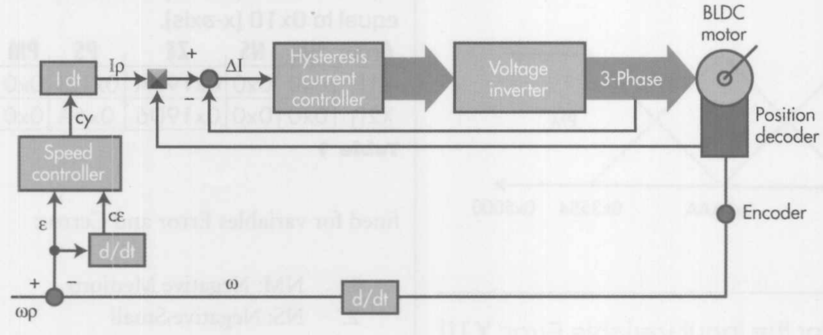


Figure 1

trol also has a shorter development cycle compared to PID controllers, and thus a faster time-to-market. This article discusses the process of using FL algorithms to control BLDC motors using a Texas Instruments c28xx fixed-point family of DSPs.

BLDC CONTROL MODEL DEVELOPMENT

Before constructing the FL engine, we must first develop a model to base the design on. FL controllers use heuristic knowledge and express the design using a linguistic description of the model. Rather than develop a model from scratch, we'll use the PID controller model as a starting point. Once developed and implemented, the FL controller is improved by adjusting its parameters.

In general, there are three design steps for developing a FL BLDC controller:

1. Define inputs, outputs, and the controller's range of operation.
2. Define fuzzy membership set functions and rules.
3. Tune the engine.

Figure 1 shows the block diagram of the BLDC controller model.

The first step is to define the relevant inputs and outputs of the model. The inputs are the error (E), which is the current error between the set speed (SS) and the current speed (CS). The other input is the change in error (CE), which is the difference between the current error, and the previously calculated error (PE). The output is the change in armature voltage (CV), which is the difference between the current armature voltage (CAV) and the stored value of the previous armature voltage (PAV). The resulting model equations are as follows:

$$E = SS - CS$$
$$CE = E - PE$$
$$CV = CAV - PAV$$

Motor speed units are in revolutions per minute (RPM), and E determines how close we are to the target speed. So for $E > 0$ motor speed is below set speed. Alternatively, $E < 0$ indicates that the motor is spinning faster than the set speed. CE determines controller direction to adjust. CE is positive if and only if (iff) the current speed is less than the set speed.

Alternatively, CE is negative iff the current speed is greater than the set speed. When close to the set speed CE alternates between positive and negative values. CV is the energizing voltage applied to the armature. This voltage is expressed in implementation as a pulse width modulation (PWM) duty cycle.

The next step is to define the fuzzy membership set functions, variables and rules. In order to work, non-fuzzy (crisp) inputs and outputs must be converted into fuzzy ones. Conversion is performed by using linguistic variables to represent input and output ranges. These are also referred to as *fuzzy variables*. Fuzzy variables are used to partition a region of values for membership functions. For example, five variables are used to map the inputs and output. They are negative medium (NM), negative small (NS), zero (Z), positive small (PS), and positive medium (PM). The model's inputs and outputs are membership set functions that are described over the range of operation by the five fuzzy variables.

Instead of math formulas, an FL controller uses fuzzy rules to make a decision and generate an output; how cool is that? FL rules are in the form of IF-THEN statements.

Fuzzy rules determine system behavior, rather than complex math equations. For example, IF the error (E) is equal to NM and change in error (CE) is equal to PS, then the change in the armature voltage (CV) is equal to NS.

The number of rules used is based on the experience of the designer and the knowledge of the system. Thus, for our system the number of rules used is 25, which is based on our basic PID controller model using the PID's control surface.

In order to energize the armature, the CV fuzzy output must be converted back to a crisp output. This process is called *defuzzification*. A popular method of defuzzification called the *center of gravity* method is used; I'll discuss it in greater detail later.

The last step of design is to adjust the membership functions and rules. This stage is also referred to as *tuning*. Tuning is used to improve the performance of the FL controller. Once designed, the controller is ready to be implemented.

The FL controller implementation is made up of three modules. They are fuzzification, rule-base, and defuzzification. The following sections discuss the modules as related to the FL-BLDC implementation.

FUZZIFICATION

Fuzzificaton is the process of converting crisp value data into fuzzy data. The resulting fuzzy data conversion is based on the degree of fuzzy set membership

The membership function for the input variables Error, and Cerror.

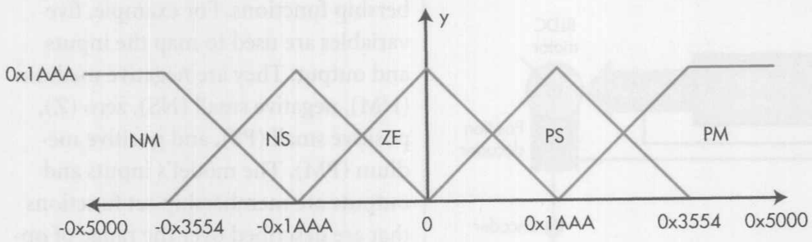


Figure 2

The fuzzified membership function for the input variable Error: X1[].

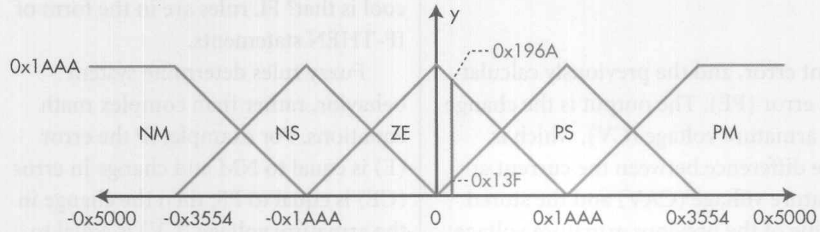


Figure 3

The fuzzified membership function for the input variable Cerror: X2[].

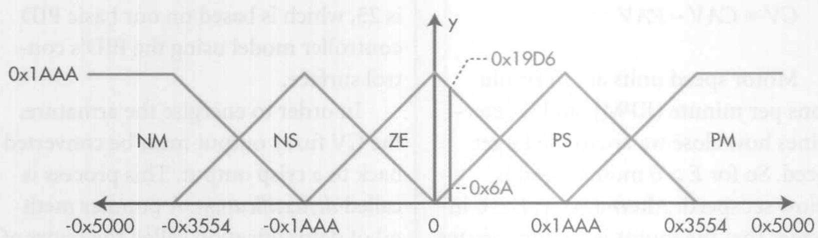


Figure 4

Fuzzified data resulting from Error equal to 0x30 (x-axis) and Cerror equal to 0x10 (x-axis).

Array	NM	NS	ZE	PS	PM
X1[]	0x0	0x0	0x196A	0x13F	0x0
X2[]	0x0	0x0	0x19D6	0x6A	0x0

Table 1

The inference rule table.

		Cerror:X2[]				
		NM	**NS**	**ZE**	**PS**	**PM**
	NM	NM	NM	NM	NS	ZE
	NS	NM	NM	NS	ZE	PS
Error: X1[]	**ZE**	NM	NS	ZE	PS	PM
	PS	NS	ZE	PS	PM	PM
	PM	ZE	PS	PM	PM	PM

Table 2

of input variables. For this application, motor control input variables are rotational error (Error) and change in rotational error (Cerror), which are taken from the PID controller model discussed earlier. *Error* is the absolute error from one sample time to the next. Similarly, *Cerror* is the change in Error from one sample time to the next. The formulas for each are:

Error = SetSpeed – CurrentSpeed
Cerror = Error – PreviousError

As mentioned during the design section, five membership sets are defined for variables Error and Cerror:

1. NM: Negative Medium
2. NS: Negative Small
3. ZE: Zero
4. PS: Positive Small
5. PM: Positive Medium

Figure 2 shows the membership sets for variables Error and Cerror. The membership sets are triangular-shaped and overlap to provide good response. Each set has a maximum value of 0x1AAA.

This differs from typical fuzzy logic literature, which sets the maximum range equal to one. Using a maximum value of 0x1AAA for the range reduces computational complexity. Specifically, the multiplying operation is reduced to either a series of additions or subtractions rather than converting to and from a floating point number.

The resulting fuzzification of the input variables produces a vector with five components that correspond to the fuzzy membership sets NM, NS, ZE, PS, PM. The value (y-axis) of each component represents the degree of membership for that crisp input value. The vectors containing the fuzzified values of Error and Cerror are denoted by arrays X1[] and X2[] respectively. For example, with Error equal to 0x30 (x-axis), and Cerror equal to 0x10 (x-axis), the resulting fuzzified data are shown in Table 1.

Figures 3 and 4 graphically show the resulting fuzzified values for Error and Cerror.

FUZZY INFERENCE RULES

Fuzzy inference rules operate on the fuzzified data to determine the system's behavior. Specifically, the fuzzified data is applied against the rule table. Linguistically, this is where the input data

Targeting high-performance I/O and processing applications, the Model 4207 delivers an on-time solution for your real-time application using the latest high-speed serial interfaces.

On board PowerPCs, FPGA, PMC/XMC sites, and extensive memory resources are connected to the industry's fastest interfaces including VXS, Gigabit Ethernet and dual Fibre Channel ports for high-speed recording.

All board resources and interfaces are joined through a configurable zero-latency crossbar switch using protocol-transparent gigabit serial data paths.

With so many connections to choose from, you are bound to find just what you need for your next system.

- Dual PMC/XMC Sites
- Freescale MPC8641D Dual Core AltiVec PowerPC Processor to 1.5 GHz
- XC4VFX60 or XC4VFX100 FPGA
- 4 GB DDR2 SDRAM
- PCI Express, Serial RapidIO™, RocketIO™
- Dual Gigabit Ethernet Ports
- Dual 4 Gbit Fibre Channel Interface
- Fabric-transparent Gigabit Crossbar Switch
- VXS, VME64x, Gigabit Ethernet-X
- Linux and VxWorks Support
- ReadyFlow® Board Support Libraries
- GateFlow® FPGA Design Resources

Call today to have it all!
201-818-5900 or go to
www.pentek.com/go/esdconnect for the 4207 brochure, technical datasheet and to request pricing.

PENTEK

Setting the Standard for Digital Signal Processing

Pentek, Inc., One Park Way, Upper Saddle River, NJ 07458
Phone: 201.818.5900 Fax: 201.818.5904 e-mail:info@pentek.com www.pentek.com
Worldwide Distribution and Support

Listing 1 Inference rules pseudo code.

```
MinValue = MaxValue = 0
Y[0..N] = 0;
For (I = 0 to N)
{
for (j = 0 to M)
{

MinValue = Min{X1[I], X2[M]};
// compare each X1 element to X2 element and store
// smaller value

// store max value found among X2 members
If (Max{X1[I], X2[M]} > MaxValue)  //
{
MaxValue = Max{X1[I], X2[M]};
}
// add max value to output vector function defined by
// membership rules
MaxValue += output_vector_member(x);
}
}
```

The output membership function.

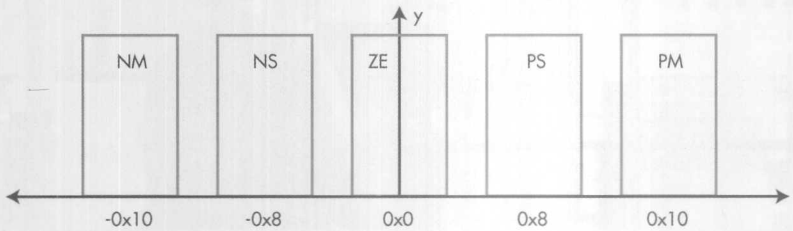


Figure 5

The output membership function multiplied by vector Y[].

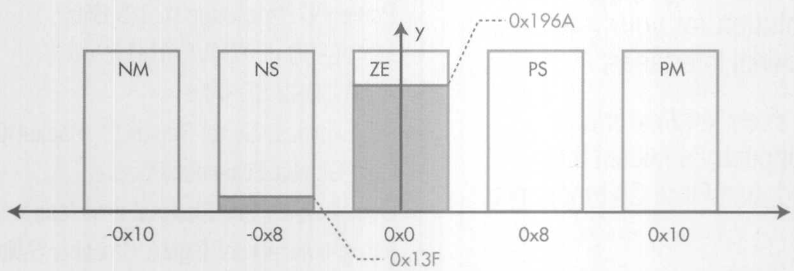


Figure 6

Error and Cerror are compared with the rule table. The rule table contains membership set components NM, NS, ZE, PS, and PM, depending on the control surface. The output is "inferred" from the valid or "fired" rules. The inference process is described by pseudo code in Listing 1.

Table 2 shows the initial rule base for the motor control surface.

In the earlier example, output vector Y[] is shown in Table 3.

DEFUZZIFICATION

Defuzzification is the process of converting fuzzy data back into crisp value data. For the purpose of this application the defuzzified value determines the duty cycle of the PWM signal used to drive the motor. The duty cycle is determined by using the modified centroid calculation function. The defuzzification approach used here takes the centroid function and multiplies it by a coefficient. The modified calculation is also known as the *centroid point calculation function*. This approach provides additional precision over the centroid calculation function.

The centroid point calculation is obtained by the center point of the function that is the result of the multiplication of the output membership function by the output vector Y[]. The formula for the centroid point calculation is:

Defuzzified Value =
Σ_$Y[i]$ X multCoeff[I] / Σ $Y[i]$

where $Y[i]$ are the i-th elements of the output vector, and multCoeff[i] are the multiplying coefficients of the output membership function. The index i has a range of $i = 1$ to $i = 5$.

Figure 5 shows the graphical representation of the output membership function used by this application with the coefficients of [-0x10, -0x8, 0x0, 0x8, 0x10].

Using the example output vector Y[] = [0x0, 0x13F, 0x196A, 0x0, 0x0], the following defuzzification output value is calculated:

```
Defuz = 0 X (-16) + 319 X (-8)
   + 6506 X (0) + 0 X (8) + 0 X
   (16) / 0 + 319 + 6506 + 0 + 0
   = -2552 / 6825 => -0.37391
```

Figure 6 shows the graphical representation of the centroid point calculation for the output vector Y[].

HARDWARE DESCRIPTION

The eZdsp2812 board is used in this motor control application. At the heart of the eZdsp board is the TMS320F2812 DSP. The TMS320F2812 is a 150-MHz device that uses timer T1 running at 20 kHz for generating PWM1-6 signals, and timer T2 running at 40 kHz for executing interrupt service routines (ISRs). Additionally, the input capture pins CAP1-3 are used to collect speed

Demonstration application motor control block diagram.

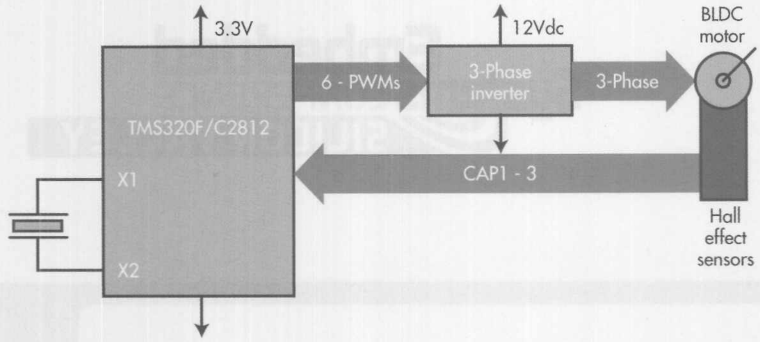


Figure 7

Motor debug session.

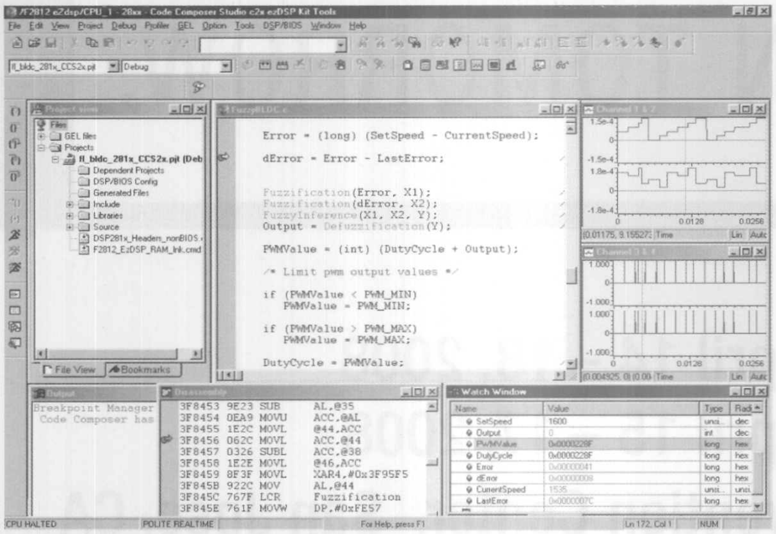


Figure 8

data from the hall-effect sensors.

Other members of the 28xx family may be substituted for the TMS320F2812. For instance, the eZdsp2808 board may be used if the timers driving the PWM and ISRs are changed. Specifically on the eZdsp2808 board, EPWM1-3 is used for PWM generation, while CPU timer 0 is used for an ISR interrupt source. Similarly, ECAP1-3 capture signals from the hall-effect sensors.

The motor is driven with PWM signals generated by the DSP and translated to a 3-phase output. The six PWM signals are used to source the 3-phase power inverter. The power inverter converts the six signals to a 3-phase signal that directly powers the motor. The 3-phase power inverter function is handled by an auxiliary motor control board. Spectrum Digital provides two boards that provide this function: the DMC550 and the DMC1500. Either board plugs directly into the eZdsp28xx board.

Hall-effect sensors are used for feedback for the fuzzy logic control loop. The commutation instants for the 3-phase power inverter switches are determined by detecting edges from signals received from the hall-effect sensors. The signals are fed into the TMS320F2812's capture pins and are debounced to eliminate noise or false edges from motor oscillations. The actual motor speed is calculated by counting the edge-triggered signals from the Hall-effect sensors via a software module. Figure 7 shows the hardware block diagram for controlling a three phase BLDC motor.

SOFTWARE DESCRIPTION

The motor control software is composed of DMC Library modules and the FL motor control routines. Seven of the DMClib modules are used in this application. They are:

- Datalog
- BLDC3PWM
- Hall3_Drv
- Mod6_Cnt
- Rmp2Cntl
- Ramp_Cntl
- Speed_PR

Additionally, the FL motor control is handled by a main `FuzzyCtl()` routine; this is `FuzzyBLDC()` for BLDC motors. When configured, these components demonstrate fuzzy logic control of a variable speed motor.

The software works by first performing configuration, then application-specific setup. Specifically, the GPIO pins are configured to act as PWMs and CAPture pins. Next the timers and module parameters are initialized, as well as ISR setup. After all peripherals are setup, interrupts are enabled, and the main control loop is entered. The main control loop calls the fuzzy controller once every 8.7 ms.

The error values are converted through fuzzification into fuzzy values and stored in X1[], and X2[]. Once converted, the fuzzified values applied to the fuzzy inference rules. The results from the inference module are stored in Y[]. Output from Y[] is then applied to the defuzzification module to convert the fuzzy value back to a crisp value. The resulting crisp value is a PWM offset that is added to the current PWM duty cycle; creating a closed loop system. The updated PWM value is checked to see if the new value is within

FUZZY LOGIC DEFINITIONS

Centroid calculation function: used for producing an exact output value by calculating the center of gravity of the union of areas bound by membership functions and the input variable axes.

Defuzzification: a general method for determining the best exact or "crisp" output of a given fuzzy set, defuzzification uses the centroid calculation function or a similar function to generate a crisp output.

Fuzzification: a method for converting a "crisp" input value to a fuzzy membership function. The resulting fuzzy value is a member of a multivalued set.

Fuzzy control system: a control system based on fuzzy IF-THEN rules that use fuzzy sets for input and output.

Fuzzy inference system: a collection of fuzzy IF-THEN rules.

Fuzzy logic: logic that uses linguistic variables to describe a system. Examples include: "fast," "slow," and "medium."

Fuzzy set: any set that allows its members to have different "grades" of membership. Each member may be expressed by a continuous number between zero and one: [0,1]. This contrasts to Boolean logic that limits set members to a value of either zero or one.

Fuzzy system: a system whose variables range over states that are fuzzy sets.

Membership function: the mapping of a fuzzy set that associates each set member with its grade of membership.

bounds, and appropriate action is taken if it is not. Finally the fuzzy controller returns the updated PWM duty cycle count to the calling routine.

Figure 8 shows a debug session of the demo application. Channels 1 &2 (the display window in the upper-right hand corner) displays the PWM counter, and the capture of the hall-effect sensors respectively.

Channels 3 & 4 (the display window beneath channels 1 & 2) show the edge-triggered versions of the PWM counter and the hall-effect sensors. The watch window displays important variables. Most important are SetSpeed, and CurrentSpeed. These values are close enough so that the output of the fuzzy controller has a zero value. Also note the other values used in the controller process. This session shows the motor under no-load conditions. The behavior may be slightly different with a load. Moreover, if a finer granularity is desired it may be necessary to tune the controller.

Byron Miller is an independent firmware engineer specializing in the design of microprocessors, DSPs, hardware debug, porting, as well as the development of firmware for control, data acquisition, fuzzy logic, and Internet appliances. He has a BA in computer science and a masters in software engineering. You can reach him at bmiller2@isd.net.

FURTHER READING:

Von Altrock, Constantin. *Fuzzy Logic and NeuroFuzzy Applications Explained*. Englewood Cliffs, NJ: Prentice Hall, 1995.

Miller, Byron. *The Design and Development of Fuzzy Logic Controllers*. Minneapolis, MN: Impatiens Publications, 1997

G. Klir, D. Schwartz, "Fuzzy logic flowers in Japan, *IEEE Spectrum* JULY 1992, p. 32.

Brubaker, David. "Fuzzy-logic basics: intuitive rules replace complex math," *EDN* June 18, 1992, p. 111.

Brubaker, David. "Fuzzy-logic system solves control problem," *EDN* June 18, 1992, p. 121.

Brubaker, David. "Design and simulate your own fuzzy setpoint controller," *EDN* January 5, 1995, p. 167.

V. Donescu, D.O. Ncacsu, G. Griva, "Design of a Fuzzy Logic Speed Controller for Brushless DC Motor Drives," *IEEE Spectrum* September 1996, p. 404.

M. Chow, Y. Tipsuwan, "Fuzzy Logic Microcontroller Implementation for DC Motor Speed Control," *IEEE Spectrum* March 1999, p. 1271.

J. Lee, T. Im, H. Sung, Y. Kim, "A Low Cost Speed Control System of Brushless DC Motor Using Fuzzy Logic," *IEEE Spectrum* April 1999, p. 433.

Miller, Byron. "A Top-Down Approach to Fuzzy Logic Design," *Embedded Systems Programming*, July 1998, p. 52.

Simon, Dan. "Fuzzy Control," *ESP* July, 2003, p. 55.

Miller, Byron. "Fuzzy Logic Does Real Time on the DSP," *Dr. Dobbs Journal*, July 2004, p. 25.

Fuzzy Logic: An Overview of the Latest Control Methodology. (TI doc – SPRA028).

Fuzzy Logic Motor Control with MSP430x14x. (TI doc-SLAA235).

TMS320F/C28xx Data Manual. (TI doc – SPRS174M).

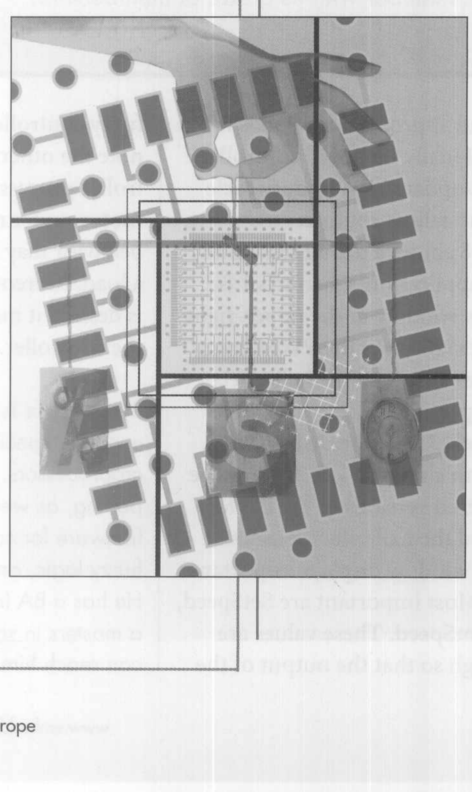
Combining a programmable solution with an industry-standard processor core can save time, money, and real estate.

Lower the cost of intelligent power control with FPGAs

BY WENDY LOCKHART

The availability of ARM's Cortex-M1 32-bit processor and single-chip, mixed-signal FPGAs make possible the development of intelligent power control that could dramatically reduce part count, board space, and system cost while increasing reliability, flexibility, and system availability. Methods exist for rapid prototyping and implementation of the hardware and software for server-based intelligent power; these methods can also be applied to a range of energy-management applications.

System considerations include development resources; tools available for development with Cortex-M1 and mixed-signal programmable system chips (PSCs); and availability of power control boards. Lowering the cost of intelligent power control requires an understanding of FPGA implementation strategies, as well as a basic



understanding of designing with ARM's Cortex-M1 microprocessor and FPGA implementation tools.

As the number of functions in a system grows, power to the system is no longer an afterthought. With time-to-market pressures and the need to support more features, designers must select peripherals from a standard set of components to meet their power and cost budgets. As a result, this may mean mixing an LCD with a 2.5-V supply and a keypad with a 1.8-V supply. With many devices, the core and I/O voltage within the device will also vary. So, within a single product, the power supply may need to generate multiple voltages and possibly different sequences of the same voltage.

In portable applications, the restrictions can be even greater, requiring power management to extend battery life. Designs can therefore become quite complex, featuring multiple supplies, controlled ramp rates, power sequencing, and complex supply management where supplies are turned on and off as needed.

Intelligent power control has existed in some high-end systems as a custom implementation or more recently with standards, such as ATCA and MicroTCA. With the market now demanding smaller, portable versions of many applications, the power control must be scaled down as well, creating a need for intelligent power control that's low cost, small form factor, and low power.

Intelligent power control involves the following basic aspects:

- Generating all the required system voltages.
- Sequencing each device's power up and down to maintain system integrity and prevent issues such as latch-up, inrush current, or I/O contention.
- Supporting the ability to switch off certain devices when not needed and power them back up for seamless operation.
- Maintaining minimum functionality in standby, with the ability to wake at certain intervals or on-demand.

Implementing these functions in an application-specific standard product (ASSP) would require a standard power profile. The use of a programmable device, however, allows for adaptation to multiple system requirements. Programmable devices also allow for future system upgrades and fine tuning. A number of programmable power modules are available, but they still need to be combined with some form of brain to tell them when to turn each supply on and off.

As shown in Figure 1, mixed-signal PSCs allow for the combination of programmable power generation with real- time analysis and control. Combining functions for power sequencing, monitoring, and control in a single chip results in significant cost savings in terms of materials list, board space, and build costs. In addition, a single device consumes less power and can manage the power supplies for the rest of the system. And, if the PSC can perform wake up from a watchdog timer or external trigger, the entire system can be shut down except for one device. Some PSCs can also manage smart battery charging, which is another intelligent way to enhance power control in the system.

Early adoption of PSCs in intelligent power control applications has been facilitated by the use of reference designs, some in the area of ATCA and others in the associated intellectual property (IP), such as Intelligent Platform Management Interface (IPMI). This platform-based approach lets designers start from a known system and customize it, or use a fixed standard and work it into their system. Some system management development kits also provide examples for a more custom approach to power control. This still requires a clear understanding of the PSC as well as a custom development environment.

Because the PSC's core includes standard flash-based FPGA gates, the Cortex-M1 core can be used as part of the design. This provides significant benefits to an existing ARM designer who must integrate intelligent power control without having to learn a

Example of a mixed-signal PSC architecture.

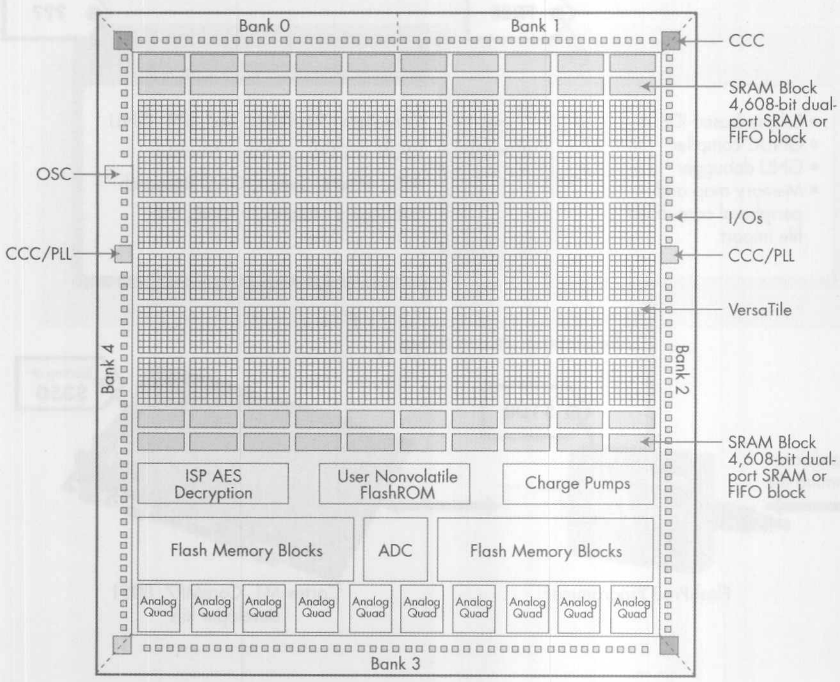


Figure 1

full suite of analog tools. Cortex-M1 is fully backwards compatible with ARM's Thumb architecture, so designers can port existing application code to the PSC.

Although implementing a Cortex-M1 design on a PSC might be a new challenge to some engineers, certain tools and techniques can make this a relatively painless design cycle. Often the easy approach from a time-to-market point of view is the expensive approach from a cost point of view. But when the design tools are created by a silicon supplier, the motivation is to help the designer get to silicon as fast as possible.

IMPLEMENTING INTELLIGENT POWER CONTROL

Standard FPGA design can take several different forms:

- Full HDL code or schematic starting from scratch.
- Combination of existing versions of design plus some new code and features.
- Combination of purchased or custom IP, existing design, and new code.

The same styles apply to both mixed-signal FPGAs and soft ARM processors on FPGAs.

The hardware and software implementation steps are shown in Figure 2. To simplify terms, the hardware implementation involves designing the processor and programming it to the FPGA. The software implementation involves developing code for the processor on the FPGA to complete the application.

First, the designer identifies available industry-standard processors (8051 or ARM, for example) for the design and then looks through offerings from each supplier to find the best list of peripherals for the design. If the perfect list of peripherals can't be found, then a CPLD or companion FPGA can be added to the board. When implementing a soft processor on an FPGA, the supplier should provide a list of available peripherals.

Ideally, the user selects the processor from a list. Then, the bus type, which is often dictated by the processor, is chosen. Once the bus is selected, the user simply drags and drops peripherals into the design. Peripherals may include interfaces such as serial peripheral interface, universal serial bus, timers, or UARTs. Unlike an ASSP processor, users can add any combination of peripherals until they run out of peripheral ports. The tool should provide the ability to auto-connect peripherals where possible and allow users to configure all other parameters and

FPGA development flow with soft processor.

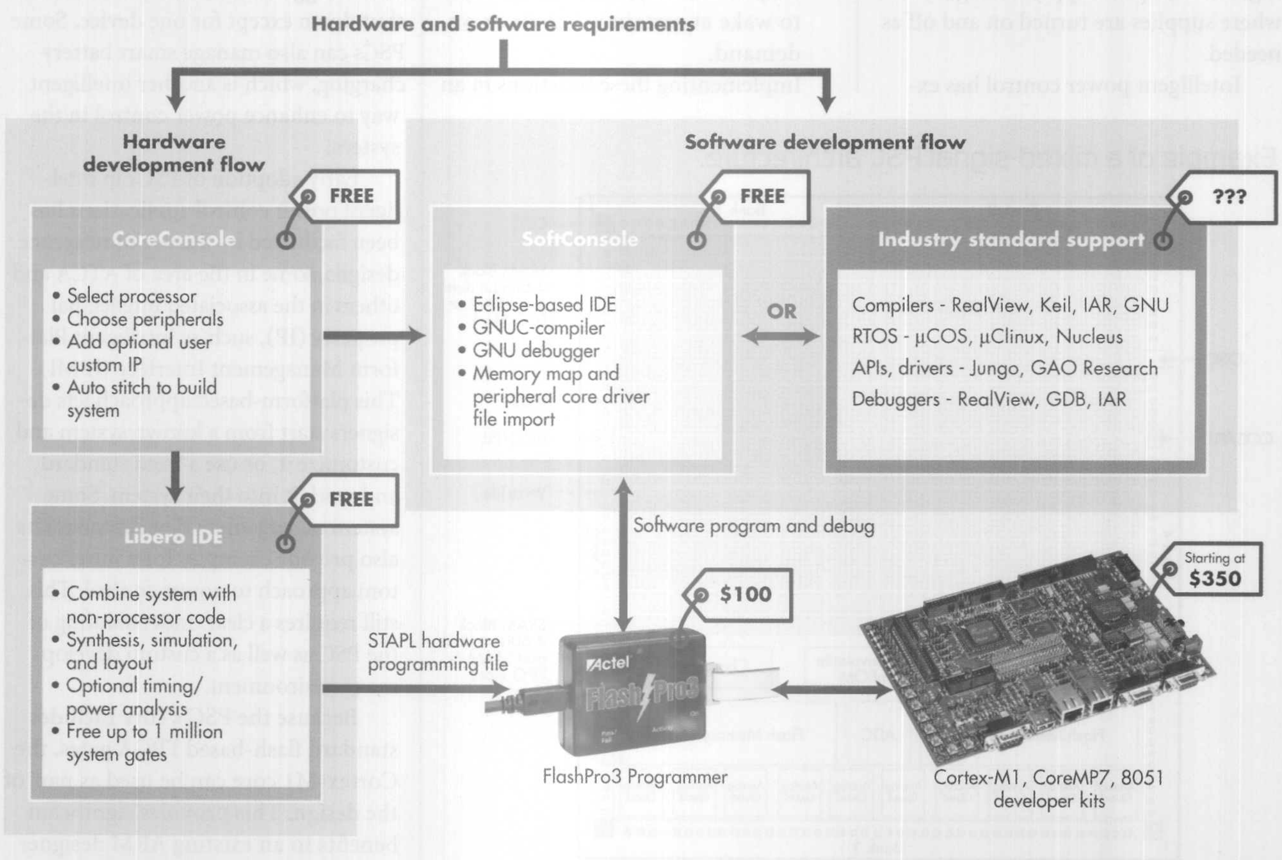


Figure 2

connections through a simple GUI.

Having designed the processor, users then generate and export to the FPGA layout tool. The processor may only be part of the FPGA design. The design may include additional glue logic or interfaces that aren't controlled by the processor. Using a simple block design tool, users can drop in the processor and any additional blocks to create a single top-level design for synthesis.

The design then follows a standard FPGA flow: synthesis, place and route, timing analysis, and so forth. Ideally, the Cortex-M1 processor should be delivered with a custom layout from the silicon vendor, indicating that timing analysis and optimization have been done. This ensures maximum performance in a minimum footprint. The layout should be customized to the device you plan to use. From here, the custom processor can be programmed to the PSC.

BOARD OPTIONS

The choice of development environment is important. Buying a "demo kit" usually means you can get the design or board to do what the supplier intended. What you really want is to convince yourself that both the technology and your design have the ability and functional accuracy to eliminate the need to build custom prototypes – or at least skip the first round of prototyping.

Look for a board that has the interfaces built-in, and has enough I/O available with connectors to work with your regular platform or peripheral board.

In addition, key components, such as crystals that can switch out, increase the board's usability as a true prototyping environment. You should also be able to program and debug on the development platform.

Software implementation is similar to a standalone processor. The tool that builds the processor should export the appropriate files required by the compiler to create the application. Existing functions can be exported and enable the ability to work from a library of functions to control the peripherals. Custom code can then be added on top of this.

For many processors, including Cortex-M1, the GNU environment offers free tools including compilers and debuggers. However, extra money does get you more mileage.

Generally, the more expensive compilers will provide either better performance or less code space. Still, free compilers will get you a functionally accurate design and a long way through prototyping. If you're working with a lot of library functions, the functions should already be optimized, eliminating the need for expensive compilers.

Debug is the next phase of design. With a soft processor, the ability to debug both the hardware and software is important. Again, a pre-optimized version of the processor helps, as processor debug won't be necessary.

Some tools are designed to coexist, such as ModelSim, which runs the hardware, and GNU, which runs the processor application. It's best to get these tools from your silicon supplier, because the PSC is unique and other debug tools won't be set up to handle the architecture. Also, with a standard ARM processor, existing ARM debug tools support debug of the processor application, but not any additional FPGA logic.

All of the methods and strategies described apply to intelligent power control, whether the application is a simple handheld portable device or for a multiserver rack system. The desire to conserve energy and cost will continue to drive electronics in this direction, while also driving designers to meet these challenges for years to come.

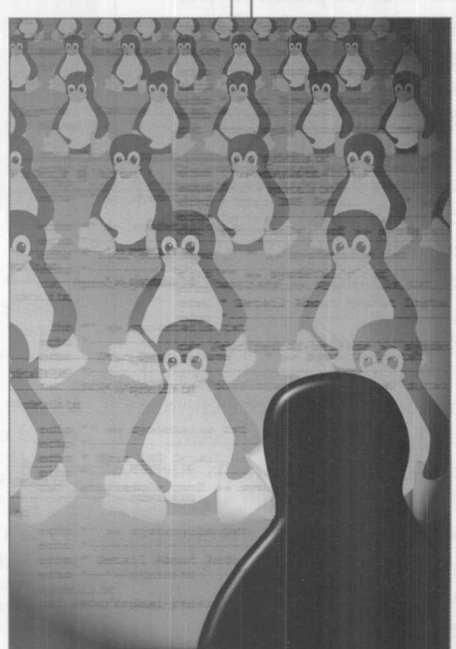
Wendy Lockhart is a principal engineer at Actel. She holds a master's degree in electronics from the University of Edinburgh, Scotland. Lockhart can be reached at wendy.lockhart@actel.com.

The system virtual machine can be used to make Linux-based applications faster and more responsive and secure. Here's a primer to get you started.

Virtualizing embedded Linux

BY GERNOT HEISER

Linux is rapidly becoming the operating system of choice in a wide array of embedded applications, ranging from mobile handsets and network/telecom infrastructure applications to media-rich consumer electronics devices such as portable media players and digital video systems.



Many embedded systems developers are already using Linux in their designs or are considering doing so. The perception among developers is that it's easier to develop applications for Linux than it is to develop for proprietary operating systems and that using Linux reduces costs because it's open source.

However, Linux still presents a number of problems in the embedded space.

First, the nature of the programming environment is often bifurcated. For example, in many media-rich consumer applications, Linux is used to run high-level application code that is similar – and often identical – to application code used on personal computers. Such code is typically developed by application programmers who normally are not experts in programming low-level embedded systems.

But such applications have much real-time functionality that requires low and predictable interrupt latency. In the case of the mobile phone terminal, the cellular communication subsystem has real-time requirements. And while embedded Linux has cer-

tainly improved, these requirements are best met by a small and highly efficient real-time operating system (RTOS).

Second is the problem of security. In a mobile-phone handset, for example, the communication stack is of critical importance – if it is subverted by an attacker, the phone could be turned into a jammer that disables communication in the whole cell. Similarly, an encryption subsystem needs to be strongly protected from being compromised.

It's no insignificant challenge to create a secure system that runs millions of lines of code; inevitably, the code contain tens of thousands of bugs, many of which can compromise the system's security. Increasingly prone to attacks, embedded Linux implementations are large enough (hundreds of thousands of lines of code) to contain as many as a thousand bugs. Because the Linux operating system normally runs in privileged mode, once it is compromised, attacks on any part of the system are possible.

Third is the issue of license separation. Linux is a frequently deployed high-level operating system. Among its advantages are the royalty-free status, independence from specific vendors, widespread deployment, and a strong and vibrant developer community.

A frequent concern about Linux is that it's distributed under the GPL license, which requires that all derived code is subject to the same license and thus becomes open source. Some legal arguments claim that the license applies even to device drivers that are loaded into the kernel as binaries at run time.

This restriction creates a potential problem for chipmakers who consider device interfaces valuable proprietary IP. An open-source device driver will effectively publish those device interfaces, a strong disincentive for using Linux in many embedded systems scenarios.

As is the trend with desktop applications developers, many embedded systems developers are looking to the

The hypervisor (or virtual-machine monitor) presents an interface that looks like hardware to the "guest" operating system.

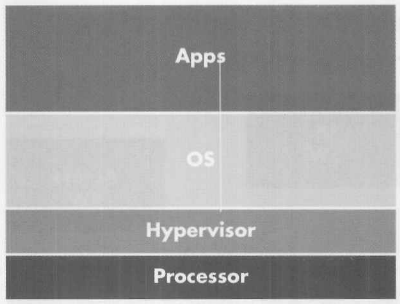


Figure 1

use of system virtualization environments, also called system virtual machines, to resolve, or at least minimize, such problems.

Unlike process virtual machine environments specific to particular programming languages, such as the Java VM, *system virtual machines* correspond to actual hardware and can execute complete operating systems in isolation from other similar instantiations in the same computing environment.

This article will explain embedded-system virtual machine models and explores where and how they can be used to make Linux-based applications faster and more responsive and secure.

THE BASICS OF VIRTUALIZATION

Virtualization refers to providing a software environment in which programs (including operating systems) can run as if on bare hardware, as Figure 1 shows. Such an environment is called a *virtual machine*. A virtual machine is an efficient, isolated duplicate of the real machine.

The software layer that provides the virtual machine environment is called the *virtual machine monitor (VMM)*, or *hypervisor*. The VMM has three essential characteristics:

1. It provides an environment for programs that is essentially identical to the original machine;
2. Programs that run in this environment show, at worst, minor decreases in speed; and
3. The VMM is in complete control of system resources.

All three characteristics are important and contribute to making virtualization highly useful. The first (similarity) ensures that software that runs on the real machine will run on the virtual machine. The second (efficiency) ensures that virtualization is practical from the performance point of view.

The efficiency feature requires that the vast majority of instructions be directly executed by the hardware: any form of emulation or interpretation replaces a single virtual-machine instruction by several instructions of the underlying hardware.

This requires that the virtual hardware be almost identical to the physical hardware on which the VMM is hosted. Small differences are possible, such as the physical hardware may miss some instructions of the virtual hardware (as long as they aren't heavily used), the memory-management unit may be different, or devices may differ.

However, not all instructions can be directly executed. The resource-control feature requires that all instructions that deal with resources must access the virtual rather than the physical resources. This means such instructions must be interpreted by the VMM, as otherwise virtualization is violated.

Specifically, the virtual machine must interpret two classes of instructions:

1. *control-sensitive instructions* modify the privileged machine state and therefore interfere with the hypervisor's control over resources; and
2. *behavior-sensitive instructions* access (read) the privileged machine state. While these instructions can't change resource allocations, they reveal the state of real resources, specifically when they differ from the virtual

resources and therefore break the illusion provided by virtualization.

BENEFITS OF VIRTUALIZATION

The key attraction of virtualization for embedded systems developers is that it supports the concurrent existence and operation of multiple operating systems on the same hardware platform.

Virtualization helps overcome the challenges caused by the bifurcated programming environments by running appropriate operating systems concurrently on the same processor core, as shown in Figure 2. The same effect could be achieved by using separate cores for the real-time and application software stacks, combined with hardware mechanisms for partitioning memory.

The ability to run several concurrent operating systems on a single processor core may reduce the bill of materials, especially for lower-end devices. It also provides a uniform operating system environment in the case of a product line (composed of high-end devices using multiple cores as well as lower-end single-core devices).

Virtualization can also be used to enhance security. A system virtual machine encapsulates a subsystem, so that its failure can't interfere with other subsystems (note that this encapsulation is courtesy of the aforementioned resource-control requirement).

TYPES OF VIRTUALIZATION

There are two basic ways to ensure that code running in the virtual machine doesn't execute any sensitive instructions: 1) *pure virtualization*, which depends on sensitive instructions not being executable by the virtual machine and 2) *para-virtualization*, where sensitive instructions are removed from the virtual machine.

Pure virtualization

The classical approach is pure virtualization, which requires that all sensitive instructions be privileged. Privileged instructions can be executed in a privileged state of the processor (typically called privileged mode, kernel mode, or supervisor mode) but generate an exception when executed in unprivileged mode (also called user mode). This is shown in Figure 3. An exception enters privileged mode at a specific address (the exception handler), which is part of the hypervisor. The hypervisor can then interpret ("virtualize") the instruction as required to maintain virtual machine state.

Until recently, pure virtualization was impossible on almost all contemporary architectures, as they all featured sensitive instructions that were not privileged. Recently many of the major processor manufacturers – including Intel and AMD in the desktop space and ARM in the embedded market – have added virtualization extensions that allow the processor to be configured in a way that forces all sensitive instructions to cause exceptions.

Despite this, there are a number of reasons this approach to virtualization is not generally used, especially in embedded applications. One is that exceptions are expensive.

On pipelined processors, an exception drains the pipeline, resulting in delay in processing, typically one cycle per pipeline stage. A similar delay typically happens when returning to user mode. Furthermore, exceptions (and exception returns) are branches that usually are not predictable by a processor's branch-prediction unit, resulting in additional latency.

These effects typically add up to 10 to 20 cycles, more in deeply pipelined high-performance processors. Add to this the work required for the actual instruction emulation, we can see that virtualizing a single instruction costs dozens of cycles. Some processors (notably the x86 family) have exception costs that are much higher than this (hundreds of cycles). This creates substantial overhead for operating system's code, which frequently executes many privileged instructions in a short time.

Para-virtualization

In this approach, the source code can be manually modified to remove direct access to privileged state and instead replace such accesses by explicit invocations of the hypervisor ("hypercalls").

Para-virtualization allows replacement of many sensitive instructions by a single hypercall, thus reducing the number of (expensive) switches between unprivileged and privileged mode.

If properly implemented, para-virtualization has the potential to reduce the virtualization overhead. Variants of para-virtualization have been de-

Virtualization allows running multiple operating systems concurrently, serving the different needs of various subsystems, such as the real-time environment vs. high-level API.

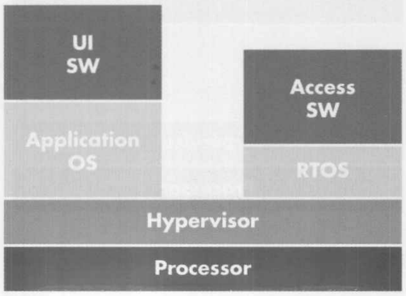


Figure 2

Most instructions are directly executed, while some cause an exception that invokes the hypervisor, which then interprets the instruction.

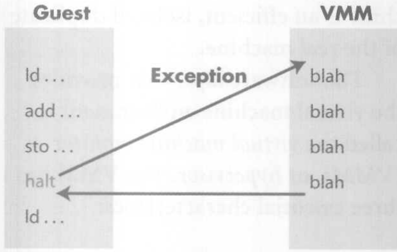


Figure 3

Structure of monolithic and microkernel-based systems.

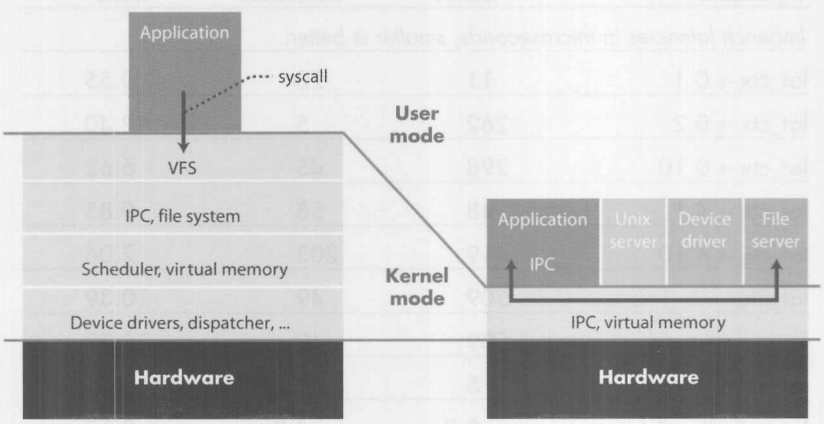


Figure 4

ployed for years by VMWare and Xen from the University of Cambridge, both aimed at the enterprise market.

Recently virtualization solutions aiming at embedded designs have emerged, such as L4/Wombat from the University of New South Wales, and the commercial systems Trango, VLX from VirtualLogix as well as OK4 from Open Kernel Labs.

All have their advantages and disadvantages in different environments, but what most of these approaches all have in common is that they introduce another layer of software – and complexity – into the operating environment. And in many embedded applications, where code size frequently measures in the millions of lines of code, breaking this into two or three virtual machines is of limited help for improving overall system reliability and security.

The isolation provided by current approaches to para-virtualization is by its nature coarse-grained – it provides the appearance of a complete machine for each subsystem. This means that each virtual machine is required to run its own operating system, making them relatively heavyweight.

Consequently, increasing the number of virtual machines in order to reduce the granularity of the subsystems would create serious performance issues and significantly increase the amount of code. This in turn not only requires increased memory size and thus power consumption, but also results in more points of failure.

USING A MICROKERNEL AS A HYPERVISOR

The most appropriate way to deploy para-virtualization in an embedded design is by integrating it into the structure of the operating system. But as Linus Torvalds commented recently, integrating it into Linux is not practical because there is no one-size-fits-all "One True Virtualization" model appropriate to all the applications in which Linux is being used. It also wouldn't help for those deploying devices without Linux (because they want to offer a choice of high-level operating systems, or maybe because they ship low-end devices in which a high-level operating systems makes no sense).

A better place to implement an integrated para-virtualization mechanism is in the companion RTOS used to handle the hard real-time operations that Linux can't. This approach requires more than a bare hypervisor: it needs a kernel that can provide basic operating-system mechanisms, such as a high-performance message-based microkernel, as Figure 4 shows.

Unlike the classical monolithic operating system structure of Linux with a vertical structure of layers, each abstracting the layers below, a microkernel-based system, exhibits a horizontal structure. System components run beside application code and are invoked by sending messages.

A notable property of a microkernel system is that as far as the kernel is concerned, there is no real difference between "system services" and "applications" – all are simply processes running in user mode. Each such user-level process is encapsulated in its own hardware address space, set up by the kernel.

A subsystem running on a microkernel can only affect other parts of the system (outside its own address spaces) by invoking kernel mechanisms, particularly IPC. It can only directly access memory (or another resource) if it's mapped into its address space via a system call.

This model is a good fit for embedded systems, where the distinction between "system services" and "applications" is frequently meaningless due to the cooperative nature of the interaction of subsystems.

The central mechanism provided by a microkernel is a message-passing communication mechanism called IPC.

In the horizontal system structure, IPC is used for invoking all system services, as well as providing other communication between subsystems. Owing to the need for high-bandwidth, low-latency communication, a microkernel typically also provides mechanisms for setting up shared memory regions between processes.

RIGHT MECHANISM

In this context, a microkernel provides the right mechanisms for efficiently supporting virtualization. The microkernel serves as the hypervisor, which catches virtualization traps. Contrary to other virtualization approaches, the microkernel forwards the exception to a user-mode virtual machine monitor, which either performs the emulation or signals a fault.

The IPC is also the enabler for low-overhead virtualization: a system-call trap executed by a guest application in a virtual machine invokes the microkernel's exception handler, which converts this event into an IPC message to the guest operating system.

The guest handles it as a normal system call. The system-call result is returned to the guest application via another IPC message, which unblocks the waiting guest process.

Similarly, IPC is used to deliver interrupts to the guest operating system's interrupt handler. It's also used to communicate with device drivers, and for communication and synchronization between any components of the system, including between virtual-machine environments.

Native vs virtualized Linux on PXA255 @ 400 MHz.

Benchmark	Native	Virtualized	Ratio
lmbench latencies in microseconds, smaller is better.			
lat_ctx -s 0 1	11	20	0.55
lat_ctx -s 0 2	262	5	52.40
lat_ctx -s 0 10	298	45	6.62
lat_ctx -s 4 1	48	58	0.83
lat_ctx -s 4 10	419	203	2.06
lat_fifo	509	49	10.39
lat_pipe	509	49	10.39
lat_unix	1,015	77	13.18
lat_syscall null	0.8	4.8	0.17

Table 1

As the same IPC mechanism is used for many different operations, it's typically highly optimized. This implicitly benefits virtualization as well as other critical system operations. As a well-designed IPC mechanism is also very simple, it's possible to optimize it in virtually all of its aspects.

A PRACTICAL APPLICATION

To see how this more fine-grained approach to virtualization has an impact on how an embedded systems designer works, let's look at a media player design, originally hosted on a more traditional para-virtualization environment with Linux as the guest operating system.

The design is then ported to run in its own address space as a native application under the open source, but commercially supported, message-based OKL4 virtualization-ready microkernel.

The media player can then run side-by-side with the Linux system (that still supports other applications) but also with a trusted crypto service that runs in a minimal trusted-computing-base environment. Over time, more components can be extracted from their monolithic environments, be it a high-level operating system or an RTOS running a communications stack, into their own protected compartments.

This includes device drivers, network stacks, file systems and other functional components. Such an approach can dramatically improve the robustness of the system by introducing internal protection boundaries that contain the damage caused by bugs.

For more than 10 years, L4 has been successfully used as a hypervisor for virtualizing Linux. As shown in Table 1, the performance of open-source OKL4-based virtual machines is typically within less than 5% of the native performance. A particularly interesting result is that of Linux on ARM platforms, where OK Linux (Linux virtualized on OKL4) outperforms native Linux in lmbench context-switching and other microbenchmarks by factors of up to 50.

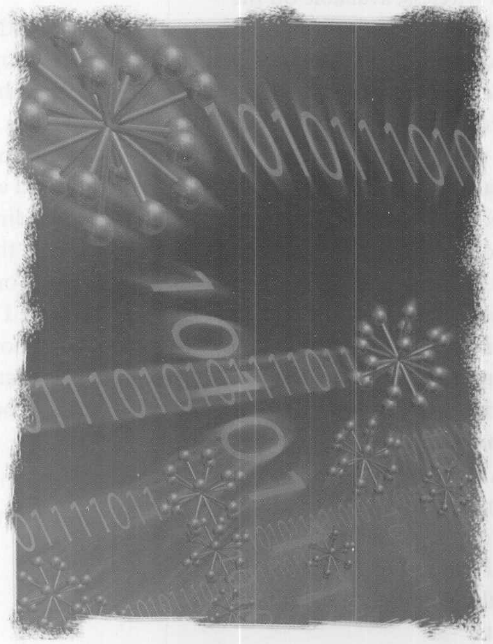
Gernot Heiser (gernot@ok-labs.com) is cofounder and chief technology officer of Open Kernel Labs (OK). Prior to founding OK, he created and led the Embedded, Real-Time and Operating Systems (ERTOS) research program at NICTA, the Australian national centre of excellence for information and communications technology. He is also a professor at the University of New South Wales. Gernot Heiser holds a PhD in computer science from ETH Zurich, Switzerland.

Back to the future: Manchester encoding

When commercial options fail, try using Manchester encoding and other time-tested protocols in low-cost, low bit-rate serial communications.

BY ROBERT GUASTELLA

When developing an embedded system that requires basic serial network communications, the list of design options are plentiful. But when all of the off-the-shelf alternatives fail to meet your requirements, it's worth remembering that traditional and well-tested protocols, such as Manchester encoding, offer some unique alternatives to the current commercial methods of wired and wireless control-data transmission.



The off-the-shelf choices are numerous. Many of today's microcontrollers offer such a wide variety of network peripherals that the possibility of not finding the right one for your application seems highly remote. Whether your system architecture requires a peer-to-peer or multi-drop topology, standards such as RS-232 and RS-485, to name a few, have loyally served the embedded development community for a long time. In addition, today's technology has expanded to include CAN bus and Ethernet as possible embedded network solutions.

For applications that are cost-sensitive, have a low data rate, and are physically constrained, some component manufacturers offer a variety of low-cost devices that interface to a simple network scheme. Dallas Semiconductor has taken it one step further and designed a family of components that communicates via

a proprietary one-wire network. This novel design provides power and communications all on a single wire. These devices work well for low-cost designs where the power required from the single wire is not significant.

In wireless network applications, numerous solutions based on the use of the ZigBee protocol atop the IEEE 802.15.4 standard for physical layer and medium access control in low-rate wireless personal area networks are available off-the-shelf from companies as diverse as Amber, Freescale, Microchip, and Texas Instruments, among others.

In addition, Microchip has an even lower bit-rate solution, combining its more sparse proprietary MiWi protocol atop the 802.15.4 physical layer. Freescale, in addition to its full ZigBee products, offers developers alternatives that use lower bit-rate protocols atop the 802.15.4 physical layer.

WHEN OFF-THE-SHELF IS NOT ENOUGH

But even with so many options, both wired and wireless, available off the shelf, there are a significant number of networked control designs where cost and physical size continue to place a heavy burden on our designs. In those cases, the usual list of suspects may not be viable after all.

Embedded developers are then challenged to dig deep and find solutions that satisfy their performance requirements, while maintaining a small physical size and a lower product cost, all without sacrificing data integrity.

For those systems requiring more power than is capable from one-wire solutions, standard off-the-shelf components may not be feasible. The solution may require a custom design that strives to balance functional requirements with a number of significant constraints.

When cost constraints are tight and performance requirements less stringent, it might be worth your while to use a "back to the future" strategy and consider the use of a well-known and well-tested serial communications protocols

A typical Non-Return to Zero (NRZ) implementation.

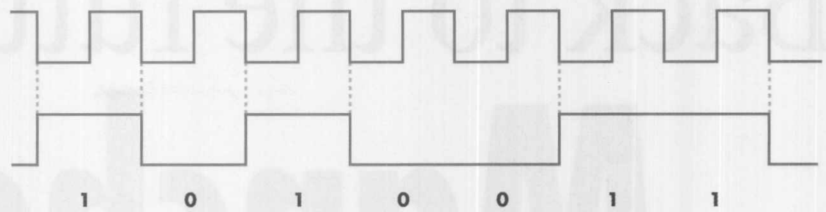


Figure 1

such as Non-Return to Zero (NRZ) or Manchester encoding as a possible solution to your embedded design.

All digital serial communications share one thing in common; they all send a series of ones and zeros. From this point, the differentiating factor becomes the level of the signal, the organization of the data, how the data is encoded, and how the data is synchronized. In a design that is sensitive to component availability and overall cost, it's highly desirable that serial data is transferred with a minimum number of wires and is built from readily available components.

NON-RETURN TO ZERO (NRZ)

One of the simplest ways to transmit digital data is by having a separate clock and data line. In this approach, a clock signal of constant frequency is synchronized with its corresponding data. Depending upon the preference of the designer, the data is either latched on the rising or falling edge of the clock. Figure 1 illustrates a typical NRZ implementation.

In this example the binary string "1010011" is being transmitted. Each bit is identified on the rising edge of the clock signal. As simple as this process is, you need to consider at least three fundamental drawbacks:

1. An additional line must support the clock signal. This is required to latch the data accurately. Depending upon the quality and length of the transmission line, additional circuitry may be required to provide the proper drive capability.
2. The second problem occurs if you decide to eliminate the clock signal. In this scenario the receiver will require an internal clock that is in near-perfect synchronization with the transmitter. Any phase shift between the transmitter and the receiver can cause bit errors to occur. This may seem a trivial matter. However, when the clock frequency becomes high enough, the sensitivity to phase differences between the transmitter and receiver becomes more significant.
3. A third limitation occurs when an NRZ transmission contains a long string of ones or zeros, as shown in Figure 2. This can make the syn-

A limitation occurs when an NRZ transmission contains a long string of one's or zeros,

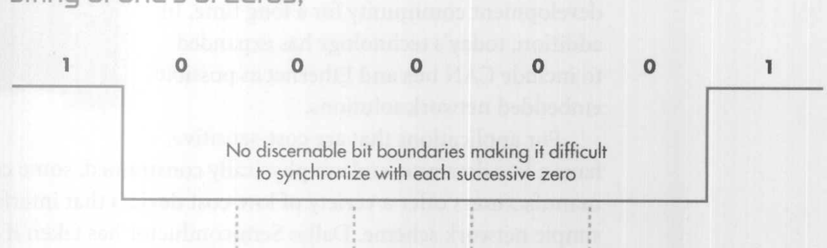


Figure 2

Two different options for encoding the data.

Figure 3

chronization between the transmitter and receiver even more sensitive to bit encoding errors. Since the line is in one state for a relatively long period of time, there are no transitions. Without transitions on the data line, it becomes impossible to see where a bit boundary is located. This can result in erroneous data encoding.

The next section describes an encoding scheme that addresses these issues and provides a unique alternative to the traditional method of data transmission.

MANCHESTER ENCODING

Manchester encoding offers distinct advantages over other digital encoding schemes. It has become a popular standard for low-cost radio frequency communication of digital data. Even Ethernet employs Manchester encoding that was used to deliver this article to your computer (if you're reading this online). So what exactly is Manchester encoding and how can it be used effectively in a low-cost embedded systems design?

Manchester encoding was first developed back in the late 1940s at the University of Manchester in Manchester, England. Given the time period and location, one with a proclivity for history might be inclined to believe its development was perhaps a by-product of research done by an obscure World War II code-breaker at England's infamous Bletchley Park.

In reality, Manchester encoding was the result of research done at the University of Manchester into phase modulation techniques used for reading and writing digital data onto a magnetic storage device. Since that time, Manchester encoding has gained wide acceptance as the modulation scheme for low-cost radio-frequency transmission of digital data.

One of the most significant characteristics of Manchester encoding is its unique way of representing digital data. Rather than representing data based on a particular level, Manchester encoding uses transitions (see Figure 3) to identify a binary one or zero. In more traditional encoding schemes, a separate clock signal determines when to sample the data line. Manchester encoding uses one signal to identify the data.

Figure 3 shows two different options for encoding the data. There has been some debate over which scheme is more advantageous. The Ethernet and IEEE standards describe Option B as the method for encoding data, while the original Manchester-encoded specification described Option A. Confusion may occur when looking at the voltage signals on a Manchester-encoded line. Most line drivers will invert their signals, which have led some to believe that Option B is being implemented. In the end, when developing your own custom-designed embedded system, the choice as to which option is best may be purely academic. However, once a choice has been made, consistent implementation is a necessity.

With this information, let's construct a Manchester-encoded bit stream using Option A. We will use the binary bit pattern shown in Figure 1. The first thing to do is establish what are called *bit boundaries*. These are the points in time where a level transition occurs. Bit boundaries are analogous to one clock period in an NRZ scheme. It's these bit boundaries that define the points where Manchester encoding of the individual bits will occur.

Figure 4 illustrates the bit stream constructed using Manchester encoding. Each point where you see an arrow is defined as the bit boundary. The arrow indicates the direction of the transition. So, using Option A from Figure 3, the first binary "1" bit is translated by a transition from a high to a low level. One clock period later, another transition occurs. This time a binary "0" bit has been encoded by a low- to high-level transition.

Further down the bit stream, you will notice a transition that looks out of place. It occurs at a point halfway between two clock periods. This is called the *setup point*. The reason for the setup point is to ensure the signal is at the correct level prior to the next bit boundary.

CONSTRUCTION OF MANCHESTER-ENCODED DATA

Manchester encoding is very easy to construct. You simply combine the se-

The bit stream constructed using Manchester encoding.

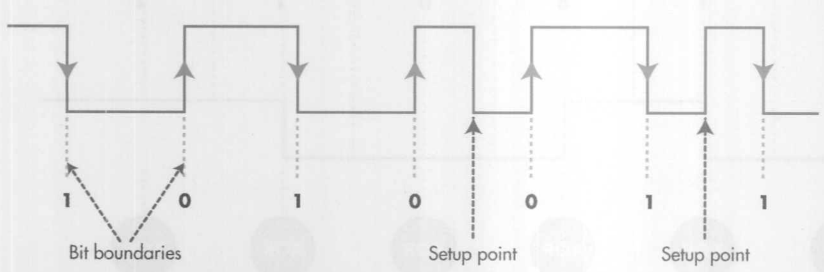


Figure 4

To create Manchester encoding, simply combine the serial bits to be encoded with the clock running at the bit-boundary rate. The Manchester-encoded output below has the same waveform as with the bit stream in Figure 4 .

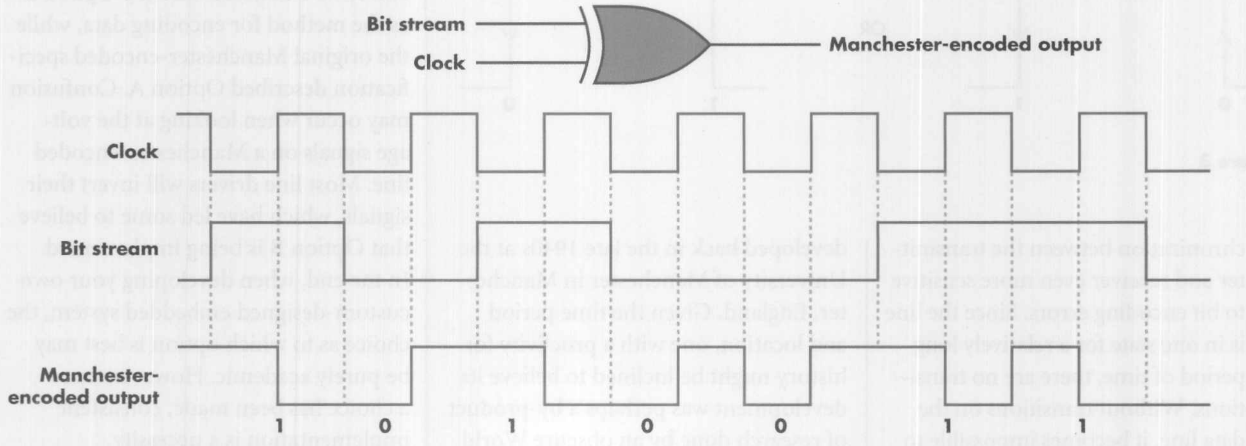


Figure 5

rial bits to be encoded with the clock running at the bit-boundary rate (see Figure 5). When you compare the Manchester-encoded output in Figure 5 with the bit stream in Figure 4, you'll see the same waveform.

DECODING MANCHESTER-ENCODED DATA

Decoding Manchester-encoded data is as easy as encoding it. You simply perform an exclusive-OR of the Manchester-encoded signal with a logical "1" at the bit-boundary sample points, as shown in Figure 6.

If you prefer an analog solution for decoding, Figure 7 shows a simple circuit that can achieve the same results. The beauty of this circuit, sometimes called a *data-slicer*, is that it doesn't require a synchronizing clock. This eliminates the possibly of errors caused by clock jitters or mismatches between the transmitting and receiving clock signals. The only issue is the values selected for the resistor and capacitor. They must be selected so that the RC-time constant is longer than ½ the bit-boundary period. This will prevent detection during a setup period. While Figure 7 represents a conceptual view, in practice you would want to create some positive feedback around the amplifier. The positive feedback will provide hysteresis that will help filter out noise that could possibly be misinterpreted as a real Manchester-encoded transition.

Once the hysteresis and the RC-time constant have been properly

To decode Manchester-encoded data, perform an exclusive-OR of the Manchester-encoded signal with a logical "1" at the bit-boundary sample points.

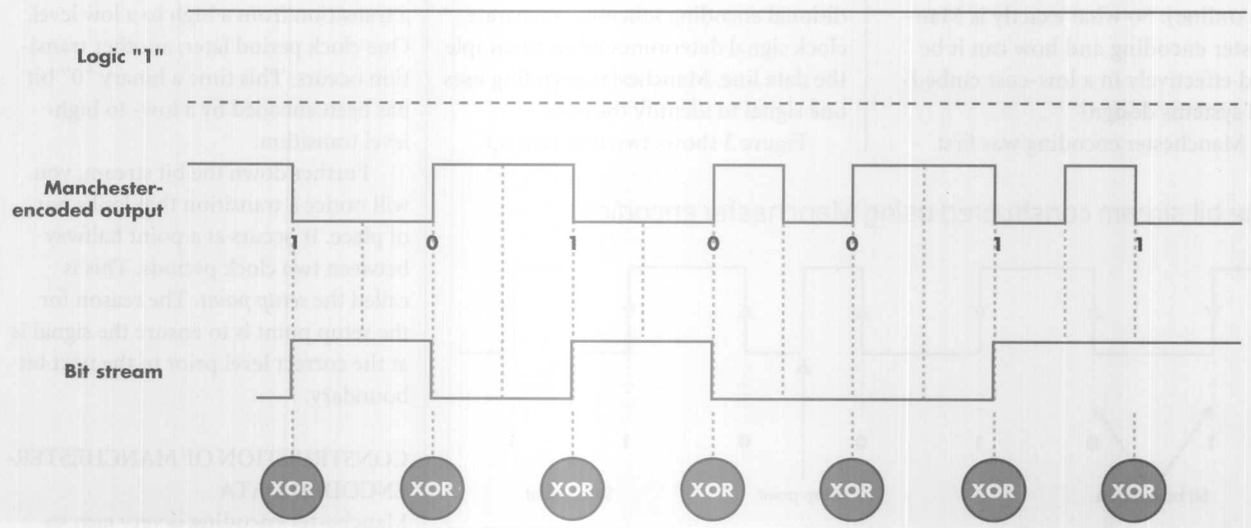


Figure 6

This analog solution for decoding shows a simple circuit that can achieve the same results as Figure 6.

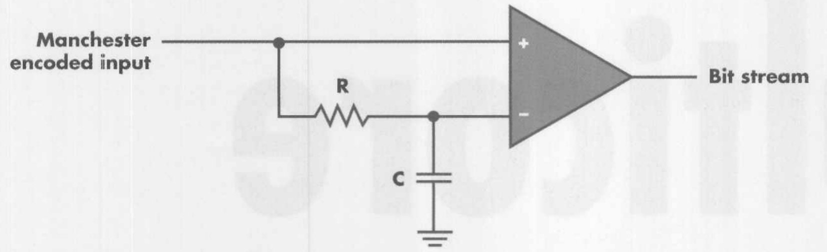


Figure 7

setup, the circuit will reliably decode Manchester-encoded signals. You will also notice that this circuit will work for both Option A and Option B.

CLOCK SYNCHRONIZATION

Another intrinsic value to Manchester encoding is the fact that the synchronizing clock is embedded within the signal. This fact is exploited in Ethernet, which uses on-board circuitry to maintain clock synchronization. A Digital Phase Locked Loop (DPLL) circuit monitors the incoming Manchester-encoded signal and makes adjustments to its internal oscillator to keep it in constant synchronization with the transmitter's clock frequency.

The DPLL functions by sampling the incoming Manchester-encoded data with its own local clock. A simple shift register, driven by the local clock, accumulates all the shifted bits.

If the local oscillator is in synchronization with the transmitter's clock, there will be an equal number of binary 1's and 0's across the shift register. If an imbalance occurs between binary 1's and 0's, the local clock is adjusted based on the number of binary bits off center. This is why you will find a preamble at the beginning of each packet transmitted via Ethernet.

Each Ethernet packet starts with an 8-byte (64 bit) preamble, which is used by the DPLL to "lock" into the correct frequency. Since the preamble doesn't contain useful data, no data is lost. However, it does add more overhead to the data stream.

A more esoteric version of Manchester encoding is a scheme called Differential Manchester encoding (DME). Think of it as Manchester encoding on steroids. DME is a more efficient encoding scheme because it requires less bandwidth than standard Manchester encoding.

The overhead of transmitting a data stream using DME is less because it doesn't require a preamble, which is used by the DPLL to lock onto the clock frequency. Because of this, DME can be found in networks, such as fast Ethernet over copper twisted-pair wiring.

DME differs from standard Manchester encoding in one simple way: Manchester encoding represents binary data based on a positive or negative edge transition at each bit boundary. DME represents data by the presence or absence of a transition between two bit boundaries. Simply stated, if a transition occurs between a bit boundary, it's represented as a binary 0. An absence of a transition signifies a binary 1.

As a complement to this reintroduction to the basics of Manchester encoding for low-bit serial network applications, a second article is available online at Embedded.com. The article will leverage from the theory presented here and offer a practical, real-world example that illustrates the simplicity of implementing Manchester encoding into a real embedded design.

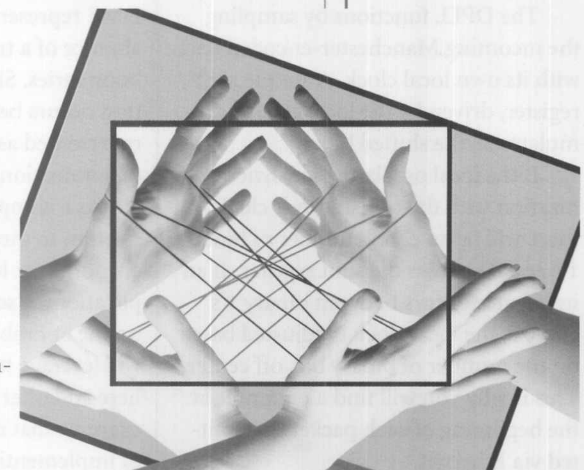
Robert Guastella (robert.guastella@tennantco) is a senior controls engineer for Tennant Company in Minneapolis, Minnesota. He has over 22 years of experience in hardware and software design on products ranging from industrial controls, to digital servo drives, to automotive electronics.

Multicore processors are here to stay but memory is a bottleneck.

Is multicore hype or reality?

BY JACK G. GANSSLE

For many years, processors and memory evolved more or less in lockstep. Early CPUs like the Z80 required a number of machine cycles to execute even a NOP instruction. At the few-megahertz clock rates then common, processor speeds nicely matched EPROM and SRAM cycle times. But for a time, memory speeds increased faster than CPU clock rates. The 8088/6 had a prefetcher to better balance fast memory to a slow processor. A very small (4 to 6 bytes) FIFO isolated the core from a bus interface unit (BIU). The BIU was free to prefetch the most-likely-needed next instruction if the core was busy doing something that didn't need bus activity. The BIU thus helped maintain a reasonable match between CPU and memory speeds.



Even by the late 1980s, processors were pretty well matched to memory. The 386, which (with the exception of floating-point instructions) has a programmer's model very much like Intel's latest high-end offerings, came out at 16 MHz. The three-cycle NOP instruction thus consumed 188 nsec, which partnered well with most zero wait-state memory devices.

But clock rates continued to increase while memory speeds started to stagnate. The 386 went to 40 MHz, and the 486 to over 100. Some of the philosophies of the reduced instruction set (RISC) movement, particularly single-clock instruction execution, were adopted by CISC vendors, further exacerbating the mismatch.

Vendors turned to Moore's Law as it became easier to add lots of transistors to processors to tame the memory bottleneck. Pipelines sucked more instructions on-chip, and extra logic executed parts of many instructions in parallel.

A single-clock 100 MHz processor consumes a word from memory every 10 nsec, but even today that's pretty speedy for RAM and impossible for flash. So on-chip cache appeared, again exploiting cheap integrated transistors. That, plus float-

ing point and a few other nifty features meant the 486's transistor budget was over four times as large as the 386.

Pentium-class processors took speeds to unparalleled extremes, before long hitting two and three gigahertz. Memory devices at 0.33 nsec are impractical for a variety of reasons, not the least of which is the intractable problem of propagating those signals between chip packages. Few users would be content with a 3-GHz processor stalled by issuing 50 wait states for each memory read or write, so cache sizes increased more.

But even on-chip, zero wait-state memory is expensive. Caches multiplied, with a small, fast L1 backed up by a slower L2 and in some cases even an L3. Yet more transistors implemented immensely complicated speculative branching algorithms, cache snooping and more, all in the interest of managing the cache and reducing inherently slow bus traffic.

And that's the situation today. Memory is much slower than processors and has been an essential bottleneck for fifteen years. Recently CPU speeds have stalled as well, limited now by power dissipation problems. As transistors switch, small inefficiencies convert a tiny bit of V_{cc} to heat. And even an idle transistor leaks microscopic amounts of current. Small losses multiplied by hundreds of millions of devices means very hot parts.

Ironically, vast numbers of the transistors on a modern processor do nothing most of the time. No more than a single line of the cache is active at any time, most of the logic to handle hundreds of different instructions stands idle till infrequently needed, and page translation units that manage gigabytes handle a single word at a time.

But those idle transistors do convert the power supply to waste heat. The "transistors are free" mantra is now stymied by power concerns. So limited memory speeds helped spawn hugely complex CPUs, but the resultant heat has curbed clock rates, formerly the biggest factor that gave us faster computers every year.

In the supercomputing world, similar dynamics were at work. Gallium arsenide logic and other exotic components drove clock rates high, and liquid cooling kept machines from burning up. But long ago, researchers recognized the futility of making much additional progress by spinning the clock-rate wheel ever higher and started building vastly parallel machines. Most today employ thousands of identical processing nodes, often based on processors used in standard desktop computers. Amazing performance comes from massively parallelizing both the problems and the hardware.

To continue performance gains, desktop CPU vendors co-opted the supercomputer model and today offer a number of astonishing multicore devices, which are just two or more standard processors assembled on a single die. A typical configuration has two CPUs, each with their own L1 cache. Both share a single L2, which connects to the outside world via a single bus. Embedded versions of these parts are available as well and share much with their desktop cousins.

Symmetric multiprocessing (SMP) has been defined in a number of different ways. I chose to call a design using multiple identical processors that share a memory bus an *SMP system*. Thus, multicore offerings from Intel, AMD, and some others are SMP devices.

SMP will yield performance improvements only (at best) insofar as a problem can be parallelized. Santa's work cannot be parallelized (unless he gives each elf a sleigh), but delivering mail-order products keeps a fleet of UPS trucks busy and efficient.

Amdahl's Law gives a sense of the benefit accrued from using multiple processors. In one form, it gives the maximum speedup as:

$$\frac{1}{f+(1-f)/n}$$

where f is the part of the computation that can't be parallelized, and n is the number of processors. With an infinite number of cores, assuming no other mitigating circumstances, Figure 1 shows (on the vertical axis) the possible speedup versus (on the horizontal axis) the percentage of the problem that can't be parallelized.

The Law is hardly engraved in stone as there are classes of problems called "embarrassingly parallel" where huge numbers of calculations can take place simultaneously. Supercomputers have long found their niche in this domain, which includes problems like weather

Possible speedup versus percentage of the problem that cannot be parallelized, assuming an infinite number of cores.

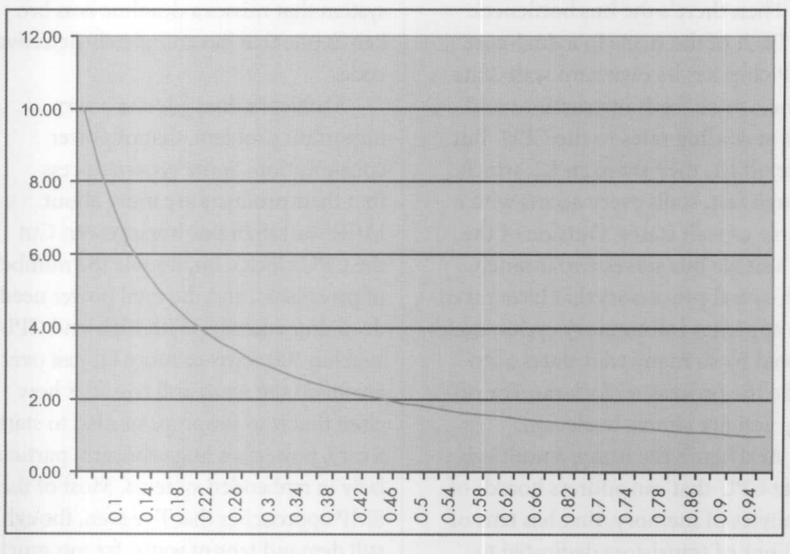


Figure 1

prediction, nuclear simulations, and the like.

The crucial question becomes: How much can your embedded application benefit from parallelization? Many problems have at least some amount of work that can take place simultaneously. But most problems have substantial interactions between components that must take place in a sequence. It's hard at best to decide at the outset, when one is selecting the processor, how much benefit we'll get from going multicore.

Marketing literature from multicore vendors suggest that a two-core system can increase system performance from 30 to 50% (for desktop apps; how that scales to embedded systems is another question entirely, one that completely depends on the application). Assuming the best case (50%) and working Amdahl's Law backwards, one sees that the vendors assume a third of the PC programs can be parallelized. That's actually a best, best case, as a PC runs many different bits of software at the same time and could simply split execution paths by application. But, pursuing this line of reasoning, assuming the dramatic 50% speed improvement comes from running one program, the Law shows that with an infinite number of processors the best one could hope for would be a 3x performance boost (excepting the special case of intrinsically parallel programs).

Then there's the bus bottleneck.

Each of the twins in a dual-core SMP chip has its own zero wait-state cache, which feeds instructions and data at sizzling rates to the CPU. But once off L1, they share an L2, which though fast, stalls every access with a couple of wait states. Outside of the L2, a single bus serves two insanely high-speed processors that have ravenous appetites for memory cycles, cycles slowed by so many wait states as to make the processor clock rate for off-chip activity almost irrelevant.

And here's the irony: a multi-gigahertz CPU that can address hoards of gigabytes of memory, that has tens of millions of transistors dedicated to speeding up operations, runs mindnumbingly fast only as long as it executes out of L1, which is typically a microscopic 32 to 64 kbytes. PIC-sized. Run a bigger program or one that uses lots of data, and the wait-state logic jumps on the brakes

A couple of Z80s might do almost as well.

In the embedded world, we have more control of our execution environment and the program itself than in a PC. Some of the RTOS vendors have come up with clever ways to exploit multicore more efficiently, such as pinning tasks to particular cores. I have seen a couple of dramatic successes with this approach.

If a task fits entirely within L1, or even spills over to L2, expect tremendous performance boosts. But it still sort of hurts ones head – and pocketbook – to constrain such a high-end CPU to such small hunks of memory.

Any program that runs on and off cache may suffer from determinism problems. What does "real time" mean when a cache miss prolongs execution time by perhaps an order of magnitude or more? Again, your mileage may vary as this is an extremely application-dependent issue, but proving a real-time system runs deterministically is hard at best. Cache, pipelines, speculative execution, and now two CPUs competing for the same slow bus all greatly complicate the issue. By definition, a hard real-time system that misses a deadline is as broken as one that has completely defective code.

Multicore does address a very important problem, that of power consumption. Some vendors stress that their products are more about MIPs/watt than raw horsepower. Cut the CPU clock a bit, double the number of processors, and the total power needs drop dramatically. With high-end CPUs sucking 100 watts or more (at just over a volt; do the math and consider how close that is to the amps needed to start a car), power is a huge concern, particularly in embedded systems. Most of the SMP approaches that I've seen, though, still demand tens of watts, far too much for many classes of embedded systems.

One wonders if a multicore approach using multiple 386s stripped of most of their fancy addressing capability and other bus management features, supported by lots of "cache," or at least fast on-board RAM, wouldn't offer a better MIPS/watt/price match, at least in the embedded space where gigantic applications are relatively rare.

Finally, the holy grail of SMP for 30 years has been an auto-parallelizing compiler, something that can take a sequential problem and divide it among many cores. Progress has been made, and much work continues. But it's still a largely unsolved problem that is being addressed in the embedded world at the operating-system level. QNX, Green Hills, and others have some very cool tools that partition tasks both statically and dynamically among cores. But expect new sorts of complex problems that make programming a multicore system challenging at best.

HERE TO STAY

Although this rant may be seen by some to be completely dismissive of multicore, that's not the case at all; my aim is to shine a little light into the marketing FUD that permeates multicore, as it does with the introduction of any new technology. Multicore processors are here to stay and do offer some important benefits. You may find some impressive performance gains by employing SMP, depending upon your specific application. Do see David Kleidermacher's article about it at *www.embedded.com/design/205203908.*

Most of my complaints disappear when each core runs code from its own memory space. Tensilica has some customers getting astonishing performance gains using this approach.

Picochip's results are impressive as well (see *www.insidedsp.com/Articles/tabid/64/articleType/ArticleView/articleId/228/Default.aspx* for a recent benchmark). But that's not SMP and is a completely different discussion.

Jack G. Ganssle (jack@ganssle.com) is a lecturer and consultant on embedded development issues.

Intel adds processors, extends life

Intel has expanded it range of devices for embedded market segments with processors with extended, 7-year life cycle support, a new chipset and a carrier-grade server. The processors, based on Intel's high-k, metal gate transistor formula and manufactured on the company's 45-nanometer (nm) process, include the Quad-Core Intel Xeon processor 5400 Series and Dual-Core Intel Xeon processor 5200 series.

These new processors, coupled with the new power-optimized Intel 5100 memory controller hub (MCH) chipset, comprise the first 45nm CPU platforms for thermally constrained bladed applications. When using the Intel 5000P chipset, the 45nm processors are suitable for full-performance and memory-intensive applications such as storage, routers, security and medical solutions, as well as communications applications such as IP multimedia subsystems (IMS).

The processors take advantage of Intel's Hafnium-based, high-k metal gate transistor formula, which reduces power consumption, increases switching speed and significantly increases transistor density over the company's previous 65nm manufacturing technology. These 45nm CPU-based platforms, based on the Intel 5100 MCH chipset, are suitable for 200 watt maximum power envelope specifications such as AdvancedTCA and also for NEBS Level-3 requirements.

Intel says it recorded a 67 percent increase it compute performance-per-watt when it validated the Intel 5100 MCH chipset-based 45nm quad-core platform. These comparisons were done using 2x Quad-Core Intel Xeon processor L5318 on an Intel 5000P chipset-based platform versus 2x Quad-Core Intel Xeon processor L5408 on an Intel 5100 chipset-based platform using the same workloads . It says that the 45nm quad-core processors also allow for a 22 percent performance gain over previous-generation platforms within the same thermal profile.

Intel is offering extended lifecycle support for 7 years for the dual-core Intel Xeon processor 5200 series (E5240, E5220, L5238) and the Quad-Core Intel Xeon processor 5400 series (E5440 and L5408). This represents an expansion from previous minimum support of 5 years.

The new Intel carrier grade server TIGH2U building block offers increased choice for customers that require power efficiency and improved compute performance for high-end communication applications. Intel also announced enhancements for its TIGW1U carrier grade server, NSW1U IP network server and NSC2U IP network server. These communication rack-mount servers now support the Quad-Core Xeon processor 5400 series and are suitable for telco and network applications in harsh environments with NEBS Level-3 requirements that demand high performance, energy efficiency and high I/O throughput.

www.intel.com

Memory controller improves power and reliability

Hyperstone has introduced the F3 family of Flash Memory controllers for solid state disk (SSD), disk-on-module (DoM), CompactFlash (CF) cards, and embedded flash. The F3 provides lower power consumption, higher reliability, endurance, and rigorous power fail safe features for all single level cell (SLC) and multi level cell (MLC) based flash memory solutions.

Based on a 32-bit RISC core including instruction set extensions optimized for flash handling, the F3 offers safe power fail handling, proven error detection and correction, superior wear leveling that involves all physical blocks including the ones containing static data, satisfying the most demanding requirements regarding data traffic and power fail situations.

It is fully compliant to CompactFlash 3.0 and compatible to 4.1 specifications and has a fast ATA supporting PIO mode 6, MDMA mode 4, UDMA mode 4 in True-IDE mode with sustained read up to 45 Mbit/s and random read up to 35 Mbit/s and sustained write up to 30 Mbit/s and random write up to 6 Mbit/s.

It provides two direct flash access (DFA) channels including two sector buffers and interleaving capability as well as error correcting code (ECC) capable of correcting 4 symbols in a 512 bytes sector with additional CRC.

As firmware is stored in flash memory, all future flashes can be supported by simple firmware upgrades while a built-in voltage regulator, and detector, reducing the bill-of-materials to a few additional capacitors and resistors.

System operation current is 75 mA max. depending on flash, automatic power-down mode, power saving mode, automatic wake-up with sleep mode current < 200 µA.

The F3 is available as known-good-die, in a 100 or 128 pin TQFP with and in either industrial or consumer temperature range.

www.hyperstone.com

Navigation module extends robust CPU cards

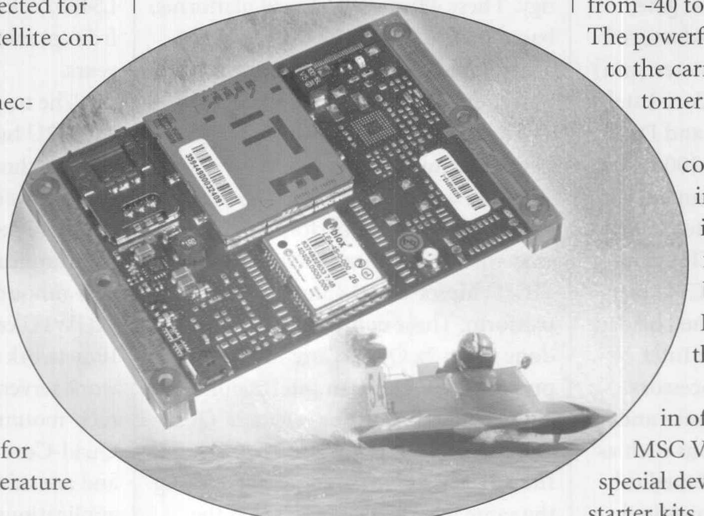
With the introduction of MSC Vertriebs GmbH's EXM32 navigation and communication module the company's CPU modules EXM32-AU1250 or EXM-SH7760 are enhanced with quad band 800/900/1800/1900 MHz EGSM and GPRS as well as with GPS functions. UMTS/HSDPA/EDGE is supported optionally. An external gyro sensor module can be connected for inertial navigation when satellite contact is not available.

Thanks to a special connector technology, the compact EXM-NAVCOM module, with dimensions of only 65 mm x 90 mm, can be easily mounted on the CPU board. Therefore, powerful navigation, telematics and fleet management systems can be implemented with very little effort.

The module is designed for an extended operating temperature range from -30 to +85°C.

In combination with a suitable GPS/GSM dual antenna, the 16 channel GPS receiver on the EXM-NAVCOM module provides the current position with an accuracy of 2.5 meters (CEP). In steady state condition the calculation of a new position takes less than 1 second, while with good signal strength the "cold start" takes less than 34 seconds. A reliable voice communication is possible via the implemented I2S audio interface. An integrated socket is provided for a SIM card.

The EXM32-AU1250 CPU module from MSC Vertriebs GmbH, designed especially for harsh industrial environment, offers good processing performance and multimedia features with low power consumption.



The device complies with the EXM32 system-on-module (SOM) standard, developed by MSC. The EXM32-AU1250 is based on the RMI Alchemy Au1250 RISC-CPU with 16 kbyte data and 16 kbyte instruction cache. It is available in two versions. In the commercial variant, specified for an ambient temperature range from 0 to 70 °C, the processor operates at a clock frequency of 600MHz. The variant for industrial ambient temperature range from -40 to +85°C operates at 500MHz. The powerful CPU module is attached to the carrier board using stable elastomeric connectors.

For applications where computing power and floating-point support are more important than multimedia features, the EXM-SH7760, based on Renesas' SH7760 RISC CPU, is available from the same module family.

For evaluation and design-in of the EXM32 modules, MSC Vertriebs GmbH provides special development platforms and starter kits.

The manufacturer offers support for developing custom-specific carrier boards. In addition to EXM32 modules, the appropriate antenna kit and compatible gyro sensor module are also available from MSC.

www.msc-ge.com

RoweBots upgrades Linux and POSIX compatible RTOS

RoweBots Research Inc. has launched Unison Version 4 for the 32 bit PIC32 micro-controller (MCU) family from Microchip Technology.

Unison is equivalent to a tiny embedded Linux RTOS which provides complete indemnification. Unison provides a very small open source POSIX RTOS which aims at increasing System on Chip (SoC) embedded development productivity and reliability.

The Unison Version 4 completes the line of offerings for all Microchips processors from Microchip's PIC24 16 bit MCUs through the dsPIC 30/33 DSCs to the PIC32 MCUs. Unison and DSPnano offer identical features and seamless support with including: integrated SoC DSP RTOS with full POSIX capabilities and a tiny foot print to minimize training time and processor size, DSP libraries with 150 functions for off the shelf tried and proven processing, complete I/O minimizing development and integration, free development, seamless integration with Microchip's MPLAB IDE for C instruction level simulation, compiling and debugging, and seamless migration between products without code changes.

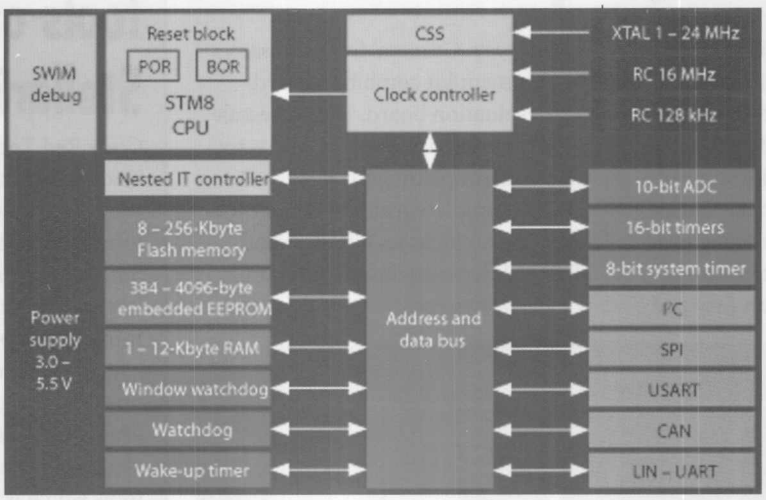
Unison V4 is hosted on Windows XP and Vista, for x86 platforms, and will begin shipping immediately.

V4 offers seamless upgrading from DSPnano running on PIC24 and dsPIC 30/33 to PIC32 without application changes.

The Freescale Coldfire V2 processor is also another SoC solution that is being supported. Like the other DSPnano chip support packages, Unison chip support will include a full complement of peripherals to eliminate low level programming.

www.rowebots.com

ST introduces 8-bit microcontroller platform



STMicroelectronics has released details of a 8-bit microcontroller platform. Combining an 8-bit core and a set of peripherals, the STM8 platform will be manufactured using an ST-proprietary 130nm embedded non-volatile memory technology.

The STM8 platform will be the foundation for a number of distinct product families each optimized for specific application areas, including automotive, industrial, low voltage and battery-operated applications, as well as application-specific standard products.

The STM8 leverages a Harvard architecture with 16-bit index registers and stack pointer, a 16Mbyte linear address space, advanced addressing modes and other features designed to optimally support C-programming to deliver leading-edge CPU performance in both speed and code density. The core reaches an average of 1.6 cycles per instruction with 20 MIPS of peak performance at 24MHz using a 3-stage pipeline.

The STM8 platform offers real embedded EEPROM with a performance in endurance and retention comparable to those of external components, making complex flash-based emulation strategies obsolete. The available on-chip flash program memory sizes will range up to 256K.

The STM8 peripherals are homogeneous across ST's 32-bit µC families such as the STM32 range, making the STM8 a perfect complement to ST's 32-bit MCU portfolio. ST will introduce the first STM8 family in the first half of 2008.

■ STMicroelectronics has made the NicheLit TCP/IP stack available free of charge with its STR91x 32-bit flash microcontrollers for networking applications. ST and InterNiche have tailored the standard stack for optimal performance with the STR91x. The stack capabilities can be extended with add-on modules already available from InterNiche.

As a fully-featured TCP/IP stack with a memory footprint of less than 12 kbytes, NicheLite includes a compact TCP layer optimized for simple memory management, data copy prevention, low memory usage and high data throughput. The NicheLite IP layer supports one hardware interface and is compatible with Internet, email and network-management protocols also available from InterNiche.

These include NicheStack PPP, FTP, Telnet server, HTTP server, DHCP server, email, and SNMP. Other protocols supported include address resolution protocol, internet control message protocol, user datagram protocol and client bootstrap protocol.

Customers developing applications for the STR91x family can implement NicheLite at no charge, with the option to purchase support from InterNiche as required. Included with the stack is the NicheTool debugging and tuning software suite, which provides powerful porting and optimization capabilities.
www.st.com

Green Hills to support OMAP35x and DaVinci

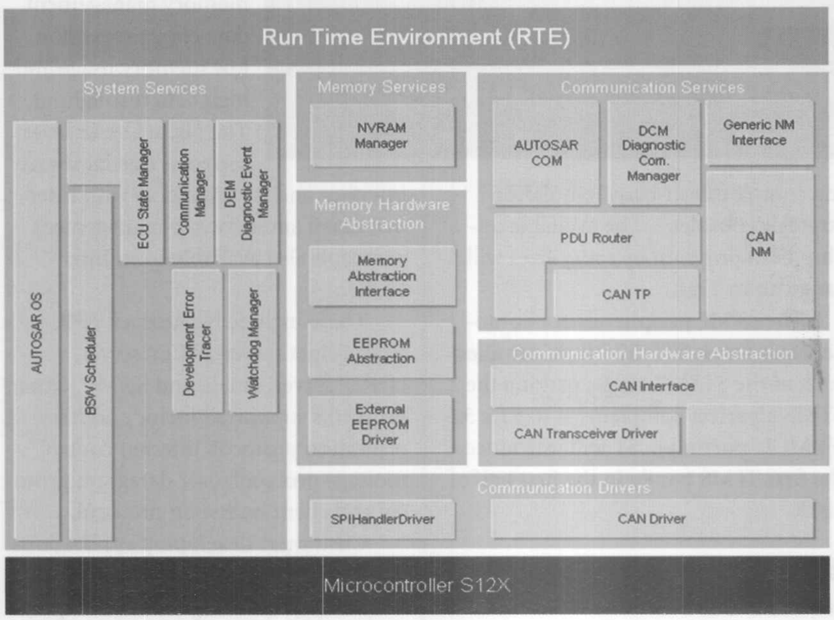
Green Hills Software, has announced plans to support the Texas Instruments (TI) OMAP35x superscalar applications processors. The Green Hills solution for the OMAP3503 consists of MULTI integrated development environment; compilers that claim 15% better code performance and density than ARM compilers; DoubleCheck static code analyzer; TimeMachine debugger, which enables forward-and-reverse code execution; and a fast simulator.

The Green Hills tools are notable for their OS-aware development and debugging. The tools have built-in support for Green Hill's OSes, as well as for Linux. Green Hills also provides extensive multi-core development support, a factor that will valuable as TI rolls out DSP+ARM OMAP processors.

Green Hills is also supporting the Texas Instruments (TI) DM355 DaVinci digital media processor. Green Hills Software's family of operating systems, optimizing C/C++ compilers, advanced debugger, MULTI IDE, and DoubleCheck are available today for DM355. TimeMachine is available today for the integrated ARM instruction set simulator and supports the embedded trace buffer on the DM355.
www.ghs.com

Autosar development kit

Geensys SA (Boulogne-Billancourt, France) has launched its first in a planned series of Autosar development kits at a price of €25,000. It comes with five days onsite support and one year's maintenance. The kit is said to enable automotive engineers to learn about and practice Autosar software development in an easy and flexible way. Geensys' first Autosar development kit combines a hardware evaluation board, with Freescale S12X microcontroller, basic software modules compliant with Autosar 2.1 and the complete Autosar Builder tool chain It also includes CANbus and NVRAM demonstration examples.
www.geensys.com



Real-time development support

Infineon Technologies AG (Munich, Germany) has announced a development support kit with a CANopen development environment for its 16-bit XE166 family of real-time signal controllers for about $75

The UConnect-CAN development kit enables the design of networked real-time motor control systems for industrial applications, such as servo-drives, air-conditioning systems, stepper motors and pumps. The Uconnect kit allows analysis of the CANopen protocol and application code for motor control designs. The kit is supplied on a USB memory stick.

The UConnect USB stick provides evaluation capabilities for the 16-bit XE164 microcontroller. The kit also contains an extension board for expanding the possible interfaces such as synchronous and asynchronous serial interfaces, six PWM channels, four AD channels and a second CAN interface.

Also included in the kit is the CANopen protocol stack from Port GmbH, a one-year license-free compiler from Altium as well as a software configuration tool, along with all documentation.

The development kit UConnect-CAN is due to become available in the second quarter of 2008 and can be ordered from Infineon sales offices and distributors for 49 euros (about $75).
www.infineon.com

Development tools offered for Stellaris MCUs

Code Red Technologies Ltd. (Cambridge, England) a provider of software development tools for 32-bit microcontrollers has ported its Code Suite development tools to the Stellaris microcontroller range from Luminary Micro (Austin, Texas).

The two companies also announced the availability of Stellaris evaluation kits featuring an evaluation version of Code Suite development tools which is always unrestricted when used with the evaluation board.

Code Suite is bult on top of the Eclipse integrated development environment with enhancements added by Code Red. Code Suite also works with real-time operating systems for designs where the resources of a real-time kernel are required, and includes a FreeRTOS.orgT project wizard.

Code Suite incorporates Code Trace, a software feature that capitalizes on the serial wire viewing (SWV) technology in Stellaris microcontrollers to deliver unprecedented visibility during debug. Code Trace delivers profiling, interrupt statistics, interrupt tracing, RTOS event tracing, and a host-strings viewing mechanism which provides printf style debugging and logging without intrusiveness. Interrupt tracing provides graphical visualization of what is occurring in a real time system.

Evaluation kits are available from Luminary Micro and Code Red Technologies that incorporate Code Suite, and allow the use of the full unrestricted use of the development environment. These kits, which are priced from $49 to $89, are available for a number of different configurations. A full license to Code Suite development tools is available through an online upgrade of the evaluation kits from the Code Red Technologies website.
www.code-red-tech.com/

Embedded Systems Design europe

MEDIA KIT AVAILABLE AT
www.embedded.com/europe

ADVERTISING SALES

EUROPE

Belgium
Nadia Liefsoens
Tel: +32-11-224 397
n.liefsoens@fivemedia.be

France, Suisse Romande
Spain & Portugal
Gérard Lecoeur
Tel: +33-4-66 32 61 06
aidmedia@wanadoo.fr

Germany PLZ 0-3, 60-65, 8-9, Austria
Victoria & Norbert Hufmann
Tel +49 911 93 97 64 42
sales@hufmann.info

Germany PLZ 4-5
Armin Wezel
Tel +49 9568 897 097
armin@eurokom-media.de

Germany PLZ66-69, 7
Helle Pedersen
Tel: +49 7131 2040396
helle.pedersen@t-online.de

Israel
Leslie Rose
Tel: +972-54-5746990
info@elarco-intl.com

Italy
Ferruccio Silvera
Tel: +39-02-284 6716
info@silvera.it

The Netherlands
Mary Zaaijer
Tel: +31-30-2291 799
mzb@mzb-mr.demon.nl

Scandinavia
Jeff Draycott
Mobile: +46-(0)702 067636
jeff.draycott@conhem.se

Switzerland
Monika Ailinger
Tel: +41-41-850 4424
m.ailinger@marcomedia.ch

UK & Ireland
Steve Regnier & Tim Anstee
Tel: +44-1732-366555
sales@starmediaservices.co.uk

U.S.A.

West, South West
Todd A. Bria
Tel: +1-831-477 2075
tbria@globalmediasales.com

Midwest
Steve Priessman
Tel: +1-630-420 8744
priessman@stevenpriessman.com

East
Jim Lees
Tel: +1-610-626 0540
jim@leesmedia.com

CANADA

West
Todd A. Bria
Tel: +1-805-226 8298
tbria@globalmediasales.com

East & Central
Steve Priessman
Tel: +1-630-420 8744
priessman@stevenpriessman.com

ASIA

Mainland China

Bejing
Boyu International Advertising Co., Ltd.
Tel: (86 - 10) 6505 9911
mdong@globalsources.com

Guangzhou
Weimei International Advertising Co., Ltd.
Tel: (86 - 20) 3879 1454 (Extension: 105)
sfung@globalsources.com

Shanghai
Lianchuang International Advertising Co., Ltd.
Tel: (86 - 21) 5306 8968 (Extension: 2131)
mandypan@globalsources.com

Shenzhen
Shenzhen ASM Advertising Co., Ltd.
Tel: (86 - 755) 8828 2536 (Extension: 2536)
bzhu@globalsources.com

Hong Kong
Asian Sources Publications Ltd.
Tel: (852) 2831 2779
tfan@globalsources.com

Malaysia
Promoexport Advertising Consultants Sdn Bhd.
Tel: (60 - 3) 5634 8500
dong@globalsources.com

Singapore
Export Advertising Consultants (Pte) Ltd.
Tel: (65) 6547 2608
ateo@globalsources.com

South Korea
Export Media Ltd.
Tel: (82 - 2) 3473 9200 (Extension: 201)
rflanders@globalsources.com

Taiwan
Asia Pacific Media Co. Ltd.
Tel: (886 - 2) 2712 6877 (Extension: 370)
ktsai@globalsources.com

Japan
E2 Publishing Co
Tel: +81-3-3556-3841
sales@eetimes.jp